

**TEACHING ROBOTS ABOUT HUMAN ENVIRONMENTS: LEVERAGING
HUMAN INTERACTION TO EFFICIENTLY LEARN AND USE
MULTISENSORY OBJECT AFFORDANCES**

A Dissertation
Presented to
The Academic Faculty

By

Vivian Chu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Georgia Institute of Technology

Georgia Institute of Technology

May 2018

Copyright © Vivian Chu 2018

**TEACHING ROBOTS ABOUT HUMAN ENVIRONMENTS: LEVERAGING
HUMAN INTERACTION TO EFFICIENTLY LEARN AND USE
MULTISENSORY OBJECT AFFORDANCES**

Approved by:

Dr. Andrea L. Thomaz, Advisor
Department of Electrical and Computer Engineering
University of Texas at Austin

Dr. Sonia Chernova, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Henrik I. Christensen
Department of Computer Science and Engineering
University of California, San Diego

Dr. Charles C. Kemp
School of Biomedical Engineering
Georgia Institute of Technology

Dr. Siddhartha Srinivasa
School of Computer Science and Engineering
University of Washington

Date Approved: January 4, 2018

What's the use, you said, of a robot that was not designed for any job? Now I ask
you-what's the use of a robot designed for only one job?

Isaac Asimov

ACKNOWLEDGEMENTS

This journey could not have been possible without the people who have helped me get to where I am today. This section merely scratches the surface of my gratitude to my advisors, mentors, colleagues, friends, and family.

I'd like to start by thanking my advisors: Andrea Thomaz and Sonia Chernova. From the moment I met Andrea, I was amazed by her ability to translate my research interests into concrete goals. On the days that I was lost or confused about what to do next, I could confidently walk into Andrea's office and know that, when I walked out, everything would be right again. Andrea's excitement about my research was also invaluable throughout my time at Georgia Tech and I am incredibly grateful that I had her support the entire time to pursue the exact research that I wanted to do. Life has an interesting way of throwing curve balls at you, and I was incredibly lucky to have had the opportunity to work with Sonia. Sonia has been an amazing support system through my final years at GT and I can't imagine where I'd be now if I didn't have her constant guidance. Thank you both for being there whenever I needed guidance on life and research, being my role models, and being there in my moments of panic during the roller coaster of the Ph.D.

I'd like to also thank my world-class committee members: Henrik Christensen, Charlie Kemp, and Sidd Srinivasa. I am honored to have them on my committee and am immensely grateful for the time they have spent in providing feedback on my research. Thank you Henrik Christensen for believing in me and supporting me also as director of IRIM; Charlie Kemp for always making the time to chat about my research; and Sidd Srinivasa for providing career advice and guidance.

I am incredibly lucky to have an amazing network of mentors and want to thank them for guiding me throughout my entire career. Thank you Dennis Lieu for pushing me to consider a graduate degree; Howard Ho, Yunyao Li, Huaiyu Zhu, Laura Chiticariu - my time at IBM opened my eyes to world-class research; Katherine Kuchenbecker - I am

honored to have had the chance to work with you at Penn and I literally would not be the researcher I am today if it were not for your constant mentorship; Leila Takayama - thank you for the warmth and willingness to listen whenever I get stuck in rough patches; Annie Anton - thank you for all of the emergency coffee chats as well as keeping a watchful eye on me even when things were going smoothly; Jivko Sinapov - thank you for being the inspiration for my research. I am so glad I had the opportunity to work with you.

Thank you to the NSF and ONR for the continual funding for basic research and science that make my pursuit of knowledge possible.

Next, I need to thank my research family: SIML and RAIL. Each and everyone of you (past and present) have been there for me when I have needed support and an ear to bounce ideas off of. Kalesha Bullard and Tesca Fitzgerald - I can't even begin to put into words how grateful I am to have had the two of you in my cohort. Through the ups and downs, excitement and dreariness of research, tea times and emergency coffee breaks, you two have been my rock and foundation at GT. I am lucky to have had an amazing lab mentor, Baris Akgun, who was always there to answer whatever question I had or just provide his debugging "aura". Thank you to Crystal Chao who provided down-to-earth advice and refreshing insight on whatever topic I brought up. Hae Won Park and Tapo Bhattacharjee, you two were my senior students mentors who I could turn to at all times for support - thank you. Thank you to my labmates and colleagues at GT: Justin Smith, David Kent, Sid Banerjee, Angel Daruna, Asif Muhammad Rana, Reza Ahmadzadeh, Aaron St. Clair, Andrew Silva, Weiyu Liu, Jon Balloch, Lakshmi Nair, Ari Kapusta, and Daehyung Park. Also, I want to thank my new UT lab family who welcomed me to Austin during my stay there: Alex Gutierrez, Taylor Kessler Faulkner, Ajinkya Jain, Yuchen Cui, and Akanksha Saran.

I want to thank the IRIM staff and faculty at GT. In particular, David Kernaghan, Josie Giles, Nina Climes, and Cynthia "CJ" Jordan, Jessica Celestine, Vivian Chandler, Jun Ueda, Tucker Balch, and Frank Dellaert. I am truly thankful for all of your support and

constant help in keeping things running in the background. RoboWomen, RoboGrad, CoC Grad Women, SWE - too many folks to list, but thank you for being my community at GT.

I could not have completed this journey without my GT and Penn families. Thank you Ian McMahon for being my partner-in-crime. I am incredibly lucky to have met you and call you my friend. Jennifer Hui, Christine Allen-Blanchette, Varsha Shankar - our little group of four got me through the crazy nights with snacks, laughter, and very little sleep. Thank you Lorenzo Riano for being the constant ray of optimism and always believing in my work. Brian Goldfain, Shan Tie, Laura Strickland - whether it be ramen nights or food runs, thank you all for dragging me out of lab and always listening to whatever craziness that happened. Kayla Despores, Andrew Price, Paul Drews, Amrita Gupta, Shelly Bagchi, Brittney English - thank you for being down for whatever adventures I needed to unwind from life. I'd also like to thank my Cal and high school friends, whom I cannot all list by name. In particular, I want to thank Nandita Sriram, Lei Huang, Andrea Lin, Annie Loch: you have been with me through thick and thin, the good and the really ugly.

I have been incredibly lucky to have such a supportive and amazing family. Thank you to my parents, Grace Liu and James Chu, for believing in me every step of the way and for supporting my crazy journey through graduate school even though it meant sending me across the country. Thank you to my brother, Willy Chu, for always knocking sense in me when I come up with hare-brained schemes and being there regardless of emergency.

Finally, to my wife, Cynthia Yeung: thank you does not even come close to expressing my gratitude for your love and support. You are my pillar, my best friend, my number one fan. Your belief in my ability to do amazing things has pushed me further than I could have ever imagined possible. I cannot have asked for a better person at my side and I look forward to battling whatever the next challenge may be together.

TABLE OF CONTENTS

Acknowledgments	iv
List of Tables	xiii
List of Figures	xv
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Approach	2
1.2.1 Affordances	2
1.2.2 Robot Exploration and Environmental Scaffolding	3
1.2.3 Multisensory Input	4
1.3 Thesis Overview	5
1.3.1 Thesis Statement	5
1.3.2 Contributions	5
Chapter 2: Background and Related Work	8
2.1 Robot Affordance Learning	8
2.2 Learning from Demonstration	10
2.3 Human-Guided Exploration for Affordance Learning	11

2.4	Multisensory Representations of Affordances	12
2.5	Learning Multisensory Robot Controllers	13
2.5.1	Model-free Learning	14
2.5.2	Environmental Model-based Learning	15
2.5.3	Segmentation Based Skill Learning	17
Chapter 3: Human-Guided Affordance Learning		18
3.1	Challenges in Robot Affordance Learning	18
3.2	Affordance as a Learning Problem	20
3.3	Human-Centered Approach	22
3.4	Hardware	23
3.4.1	“Curi” Robot	24
3.4.2	“Prentice” Robot	24
3.5	High-Level Pipeline	25
3.5.1	Action Generation	27
3.5.2	Data Collection: Robot Exploration	27
3.5.3	Affordance Modeling	29
3.5.4	Evaluation	31
Chapter 4: Learning Haptic Affordances from Demonstration and Human-Guided Exploration		35
4.1	Approach: Learning Haptic Affordances	36
4.2	Hardware Platform	38
4.3	Action Acquisition	38

4.4	Human-Guided Exploration	38
4.5	Learning Haptic Affordance Models	40
4.5.1	Hidden Markov Models	40
4.5.2	Training	41
4.5.3	Modeling Results	41
4.6	Affordance Testing	43
4.6.1	Experiment Setup	43
4.6.2	Results	44
4.7	Findings of Learning Haptic Affordances with Human Guidance	46
4.7.1	Action Variability	46
4.7.2	Action Generation	47
4.7.3	Contributions	47
	Chapter 5: Human-Guided Robot Self-Exploration	49
5.1	Exploration Space for Affordance Learning	50
5.1.1	Objects and Actions	51
5.1.2	Affordances	53
5.2	Baseline Exploration Strategies	54
5.2.1	Self-Exploration (SE)	55
5.2.2	Human-Supervised Exploration (HSE)	57
5.2.3	Active Exploration Baselines	59
5.3	Guided Exploration Strategies	60
5.3.1	Guided Aggregate Exploration (GAE)	61

5.3.2	Guided Iconic Exploration (GIE)	63
5.3.3	Guided Boundary Exploration (GBE)	65
5.4	Affordance Modeling	67
5.4.1	Model Representation	69
5.4.2	Training and Testing	69
5.5	Aggregate Results	69
5.5.1	Exploration Coverage	70
5.5.2	Model Performance	73
5.6	User Specific Results	75
5.6.1	Exploration Coverage	77
5.6.2	Clustering	79
5.6.3	Clustering Performance	81
5.6.4	Ratio of Success and Failure	83
5.6.5	Affordance Effect Space	87
5.6.6	Qualitative Observations	89
5.7	Findings of Human-Guided Robot Self-Exploration	92
5.7.1	Variation in Teaching	92
5.7.2	Action Generation	93
	Chapter 6: Affordance Transfer and Multisensory Input	94
6.1	Affordance Transfer	94
6.1.1	Hierarchy of Affordance	94
6.1.2	Affordance Category Breakdown	96

6.1.3	Transfer by Category	98
6.2	Multisensory Feedback	99
6.3	Contributions	99
Chapter 7: Multisensory Affordances for Adaptive Object Manipulation		101
7.1	Approach: Real-time Multisensory Affordance-based Control (RMAC) . . .	103
7.1.1	Data Collection	104
7.1.2	Segmentation	104
7.1.3	Affordance Switching Matrix	106
7.1.4	Subskill Segment Modeling	107
7.1.5	Execution	109
7.2	Offline Validation: Adapting Learned Affordance Models to Changed Ob- jects and New Objects	110
7.2.1	Experimental Setup	110
7.2.2	Data Collection	113
7.2.3	Multisensory Features	114
7.2.4	Training Sensory Models	116
7.2.5	Test Set	117
7.2.6	Results - Case 1: Drawer	118
7.2.7	Results - Case 2: Lamps	123
7.3	Online Validation: Adapting Learned Affordance Models in Real-time . . .	125
7.3.1	Real-time Controller	126
7.3.2	Results	127
7.4	Findings of Multisensory Affordances for Adaptive Object Manipulation . .	129

Chapter 8: Conclusions and Future Work	130
8.1 Human-Centered Framework for Robot Affordance Learning	130
8.2 Human-Guided Robot Self-Exploration	131
8.3 Multisensory Affordance Model	131
8.4 Adaptive Object Manipulation using Multisensory Affordances	132
8.5 Discussion and Future Work	132
8.5.1 Human Guidance	133
8.5.2 Multisensory Input	134
8.5.3 Tasks	135
8.5.4 Summary	135
Appendix A: User Study	137
Appendix B: Code base	147
B.1 HLP-R	147
B.2 Other	148
Appendix C: Data	149
References	162

LIST OF TABLES

3.1	Metric Equations	32
4.1	Affordances	39
4.2	Affordance Skill Monitoring Results	42
4.3	Scoop-able Leave One Object Out	44
4.4	Open-able Leave One Object Out	44
5.1	Affordances	54
5.2	Number Examples from Each Exploration Strategy	59
5.3	Percentage of Positive Interactions Per Strategy	70
5.4	Classification Scores on All Exploration Strategies	73
5.5	User Specific Classification Scores for Strategies HSE, GIE, and GBE . . .	76
5.6	Average Distance Between Demonstrations	78
5.7	User Specific Classification Scores for HSE, GIE, and GBE After Cluster- ing for Breadbox-Move and Pastajar-Move	84
5.8	Percentage of Positive Interactions Per User For Strategies HSE, GIE, and GBE	85
6.1	Affordance Triple Subcategories	95
7.1	Sensor Data	114

7.2	Sensory Model Combinations	116
7.3	Drawer Fully Closed Detailed Times	119
7.4	Drawer Opened 1 inch Detailed Times	120
7.5	Drawer Opened 3 inch Detailed Times	120
7.6	Drawer Opened 5 inch Detailed Times	120
7.7	Drawer Opened 7 inch Detailed Times	120
7.8	Original Lamp Detailed Times	124
7.9	New Lamp 1 Detailed Times	124
7.10	New Lamp 2 Detailed Times	125
7.11	Online Drawer Results Using Full Multisensory Model	128
7.12	Online Lamp Results Using Full Multisensory Model	128

LIST OF FIGURES

3.1	Visual representation of the components of an affordance in blue. Below it is an example affordance, open-able, broken down into the components of an affordance.	20
3.2	Affordances in relation to tasks and the importance of modeling each affordance to help achieve a task.	20
3.3	Affordances can be broken down into learning object-action pairs. This results in a supervised learning problem.	22
3.4	“Curi” robot used in several experiments. The various sensors used throughout the experiments include F/T sensing, RGB-D data, and audio.	23
3.5	“Prentice” robot used in several experiments. The various sensors used throughout the experiments include RGB-D data, gripper position, and audio.	24
3.6	High-level pipeline of our human-centered approach to learning affordances. We use KLfD to acquire actions from humans end-users. These demonstrations are used by the robot to explore the environment and collect multisensory data. This data of both successful and “near-miss” exploration is then used to model object-action pairs.	25
3.7	High-level view of contributions and overall approach to robot learning. The robot is given demonstrations from non-experts and uses these demonstrations to explore the environment and collect data from various sensors. This multisensory data is used to model the environment and the models are used to perform manipulation tasks on new objects.	25
3.8	Kinesthetic keyframed-based LfD for the open-able affordance on the pasta jar. Keyframes are recorded using verbal commands listed below each image.	26
3.9	Curi executing demonstrated trajectory on the pasta jar. (b) Curi misses the first time, (c) a person adjusts the object, and (d) Curi succeeds	28

3.10	Object-action pairs represented using Hidden Markov Models. We learn two HMMs per object-action pair to represent success and “near-miss” interactions. The effect space depends on the modality (haptic, visual, audio) for each experiment.	29
3.11	Visually representation for Hidden Markov Models as affordances. Where along the trajectory (action space) can be estimated as the hidden state, Z_n , which is represented using multivariate Gaussians of the multisensory robot data of the effects of the action.	31
3.12	Offline transfer evaluation where the open-able object-action pairs from the jar, box, and drawer are tested on the breadbox. Similarly, this can be done with the different cups for scoop-able.	33
4.1	Our experimental platform, Curi, with the various objects it learns affordances for in this chapter.	36
4.2	Components and information flow of the system. Action Acquisition builds an action trajectory from human demonstration. In Human-Guided Exploration, environmental scaffolding yields successful and “near-miss” interactions. This data is then used during Affordance Modeling to build a set of generative object-action models where in Affordance Testing, they are used to determine if an object has an affordance.	37
4.3	Scoop-able objects: Left-to-right - Bowl of macaroni, Cup 1, Cup 2, Parmesan Bottle	39
4.4	Open-able objects: Left-to-right - Pasta Jar, Drawer, Wooden Box, Bread Box	40
4.5	An example of force data we collect from the sensor during a scooping skill. The yellow shaded portion indicates the time period when the hand is in contact with the object. The top graph is a success and the bottom a “near-miss”.	41
4.6	Scoopable: Accuracy values for Leave One Object Out. The figure shows the accuracy breakdown between successful and “near-miss” interactions . . .	45
4.7	Openable: Accuracy values for Leave One Object Out. The figure shows the accuracy breakdown between successful and “near-miss” interactions . . .	46
5.1	A naïve user during the user study teaching “Curi” the robot that the drawer has the open-able affordance.	50

5.2	Shown are the various objects the robot explored. The top row are the objects before interaction and the bottom row include the same objects with the effect the robot is looking for. Note: pushing the drawer and pasta jar shift the object on the table.	52
5.3	Shown are the two primitive actions (pick and move) taught by users during the user study by using keyframe-based kinesthetic learning from demonstration while the arm is in gravity compensated mode. The users indicate poses within a keyframe using voice commands seen below each image. The pick action (5.3a,5.3b,5.3c) consists of a start, close, and end pose and is being demonstrated on the lamp. The move action (5.3d,5.3e) consists of a start and end pose and is being demonstrated on the drawer	53
5.4	Shown above is a visual example of the self-exploration algorithm. The algorithm is viewed in two-dimensions to be visually clear. The exploration is centered around the starting position of the object and the two depths of exploration are shown in two different shapes.	55
5.5	Shown is a visual example of the GAE algorithm. The algorithm is viewed in two-dimensions to be visually clear. The exploration is centered around the mean ending position of the first demonstrated by all of the users and exploration is bounded by the variance of the demonstrations.	61
5.6	Shown is a visual example of the GIE algorithm. The algorithm is viewed in two-dimensions to be visually clear. The exploration uses the first successful and first unsuccessful demonstrations to determine the resolution of exploration as well as where in the space to explore around.	64
5.7	Shown is a visual example of the GBE algorithm. The exploration uses the first successful and first unsuccessful demonstrations to determine the resolution of exploration as well as where in the space to explore around. . .	65
5.8	Shown are example interactions of Curi executing the move action on the bread box to find the open-able affordance. The top row (5.8a,5.8b,5.8c) show Curi successfully finding the open-able affordance. The bottom row (5.8d,5.8e) shows an example of Curi failing to find the affordance.	66
5.9	The action space (EEF relative to the object) for all five strategies for the affordance Breadbox-Move. Successful interactions are circles and failed interactions crosses. Note: 5.9b, 5.9d, and 5.9e are aggregates over all of the user models.	72

5.10	This shows the first successful and unsuccessful demonstration for the affordances Breadbox-Move, Drawer-Move, and Drawer-Pick. The symbols indicate if the demonstration given was a success or fail demonstration. The colors separated the users that could generate GIE/GBE models and those who could not (green - models were generated, red - models were not). Note: this figure requires color to fully understand.	77
5.11	Shown is the action space (EEF relative to the object) for strategy Guided Iconic Exploration for users 2, 6, and 8 for the object-action pair Breadbox-Move. The arrow indicates the direction the EEF palm is facing. Successful interactions are circles and failed interactions are crosses.	78
5.12	Displayed are the first success and fail demonstration from each user in action space (EEF relative to object) for different clusters sizes (2,3,4). The arrow indicates the direction the EEF palm is facing. Note: this figure will be easier to decipher with color.	80
5.13	Aggregate F_1 values across users and affordances and clusters sizes.	82
5.14	Displayed are the top three principal components of the final state of each user specific model for the object-action pair Drawer-Move. The values are separated in color by user models that performed well (green) and user models that performed poorly (red).	88
5.15	Shown are the comparisons of variance across the mean HMM values between good user models and poor performing user modes for all four object-action pairs. The means are separated by successful HMMs and unsuccessful HMMs. Poor performing user models overall have higher variance than high performing user models.	90
6.1	Affordances can be broken down into high-level and low-level affordances that relate directly to the actions and effects used to represent each affordances. High-level affordances can be broken down into several low-level affordances, which then have simple primitive actions that are directly related to the low-level affordance.	95
7.1	Robot platform, Prentice, turning on a lamp	102
7.2	Real-time Multisensory Affordance-based Control (RMAC). The robot first collects multisensory data using human-guided exploration. The data is then broken into two sets of subskill segments that are used to generate a switching matrix, an action model, and a sensory model to represent the affordance.	103

7.3	Kinesthetic demonstration using keyframes for turning on the lamp.	105
7.4	The various configurations the drawer is placed in for the robot to adapt its affordance controller. The robot pulls the drawer to fully opened from the 5 different starting configurations that change in 2 inch increments.	111
7.5	The various lamps the robot turns on in this chapter	112
7.6	Prentice executing demonstrated trajectory on the drawer	113
7.7	Computed features from one interaction with the lamp. The different sensory channels are displayed with vertical lines that indicate the location of the segments of the subskill segment set D_A	115
7.8	Computed features from one interaction with the drawer. The different sensory channels are displayed with vertical lines that indicate the location of the segments of the subskill segment set D_A	116
7.9	Opening the drawer comparison across the different modalities when the drawer is fully closed. The vertical red bars indicate where the model predicted it should stop and the blue vertical bars indicate where the ground truth stopping point is.	118
7.10	Average and standard deviation of absolute difference in stopping times for different drawer configurations across the 7 modalities.	119
7.11	Comparing turning on the New Lamp 2 across the different modalities. The vertical red bars indicate where the model predicted it should stop and the blue vertical bars indicate where the ground truth stopping point is. For <i>Haptic</i> , <i>Haptic+Visual</i> , <i>Haptic+Audio+Visual</i> , the robot stops after turning on the lamp. For <i>Audio + Visual</i> the robot never stops pulling. For <i>Audio</i> , <i>Visual</i> , <i>Haptic+Audio</i> the robot stops too early.	122
7.12	Average and standard deviation of absolute difference in stopping times for different lamps across 7 modalities.	124

SUMMARY

The real world is complex, unstructured, and contains high levels of uncertainty. Although past work shows that robots can successfully operate in situations where a single skill is needed, they will need a framework that enables them to reason and learn continuously so that they can operate effectively in human-centric environments. One framework that allows robots to aggregate a library of skills is to model the world using affordances. In this thesis, we choose to model affordances as the relationship between a robot’s actions on its environment and the effects of those actions. By modeling the world with affordances, robots can reason about what actions they need to take to achieve a goal. This thesis provides a framework that allows robots to learn affordance models through interaction and human guidance.

Within the scope of robot affordance learning, there has been a large focus on learning visual skill representations due to the difficulty of getting robots to interact with the environment. Furthermore, utilizing different modalities (*e.g.* touch and sound) introduces challenges such as different sampling rates and data resolution. This thesis addresses the above challenges by providing methods to interactively gather multisensory data using *human-guided robot self-exploration* and an approach to integrate visual, haptic, and auditory data for *adaptive object manipulation*.

We take a human-centered approach to tackling the challenge of robots operating in unstructured environments. The following are the contributions this thesis makes to the field of robot learning: (1) a *human-centered framework for robot affordance learning* that demonstrates how human teachers can guide the robot in the modeling process throughout the entire pipeline of affordance learning; (2) several novel *human-guided robot self-exploration* algorithms that use human guidance to enable robots to efficiently explore the environment and learn affordance models for a diverse range of manipulation tasks; (3) a *multisensory affordance model* that integrates visual, haptic, and audio input; and (4)

a novel control framework that allows *adaptive object manipulation using multisensory affordances*.

CHAPTER 1

INTRODUCTION

1.1 Motivation

“Co-robots: robots [...] that work beside or cooperatively with people” - National Robotics Initiative (National Science Foundation, 2014)

Advances in affordable sensing and actuation have led to a rapid drop in the price of state-of-the-art robots, thus enabling more industries to consider automation. However, as we move away from factories where robots are hidden behind closed doors, we encounter challenges that come from robots co-existing in human environments. The National Robotics Initiative (NRI) ¹ was started to address the scientific challenges of co-robots, specifically research that “emphasizes the realization of such co-robots working in symbiotic relationships with human partners” (National Science Foundation, 2014). The very existence of the NRI highlights that a knowledge gap exists in the integration of robots in human domains where the environment is highly dynamic, cluttered, and contains a large number of novel objects compared to controlled factories.

While robots have been used in factories for decades, traditional pre-programmed and hand-tuned control schemes lack robustness and break down when faced with the variability found in the real world. This is especially true of robotic applications that require dexterous manipulation (e.g. health care, flexible manufacturing) (Technology et al., 2013). Fortunately, while we lose structure in human environments, we gain the help of people. In this thesis, we show that typically impossible tasks for robots become tractable when human teachers, a rich source of information, are added. We believe that this relationship is key to successfully deploying robots into the real world to work alongside people in new

¹a multi-agency research funding program

environments such as homes, hospitals, and small factory floors.

1.2 Approach

This thesis looks at how a co-robot can accomplish tasks in unstructured environments. Consider the situation where an end-user brings their personal robot home from the store. The robot needs to learn about its new environment and connect its knowledge of the world with sensory data. While a robot can be deployed with pre-programmed controllers similar to robots on factory floors, these controllers need to adapt to the variability of the real world. To address this challenge, a robot needs to learn, model, and adapt to its environment throughout its life. Motivated by how humans tackle these challenges, this thesis advances the field of robotics by contributing computational approaches to skill adaptation and learning.

For the rest of this chapter, we describe the different methods by which we teach robots about human environments and the psychology literature that inspires and motivates each method. Specifically, Section 1.2.1 describes how we represent the world using affordances, Section 1.2.2 dives into why exploration and human guidance are crucial to robot learning, Section 1.2.3 motivates the necessity for robots to utilize multiple sensory inputs, and we conclude the chapter in Section 1.3.2 with the contributions of this thesis.

1.2.1 Affordances

We take an affordance approach to modeling the environment, whereby the robot is building representations of its actions and the effects that they have on objects in the environment. The term “affordance” was first introduced by psychologist J.J. Gibson (1977). We use the ecological definition of “action possibilities” that appear between an agent and the environment, which is commonly used in robotics (Şahin et al., 2007; Montesano et al., 2008). Affordances provide a building block for performing tasks and allow a robot to reason between *how* it can perform a skill and *why* that skill needs to be done. Affordances

leverage language to connect intentions and goals of people to the actions that the robot needs to perform to accomplish these goals.

For a more concrete example, let us use an object such as a cup with water to demonstrate this concept. If a robot were to perform the action *tilt* on the cup to produce the effect of water pouring from the cup, the cup has the affordance “pour-able”. By teaching a robot about this affordance, as well as other affordances (*e.g.* contain-able, open-able, etc.), the robot can build a library of building blocks throughout its life to complete tasks such as “watering the plant”. Furthermore, the affordance representation is convenient for robot learning because it connects the physical actions the robot must perform to the actual effects that robots should expect. By connecting physical motor commands and sensory inputs to task goals, robots can reason about what actions they need to take to achieve these tasks and what sensory inputs should be used to verify completion of a task.

1.2.2 Robot Exploration and Environmental Scaffolding

Co-robots deployed in unstructured human environments will have to learn and model affordances for their specific environment quickly and with minimal effort by human end-users. For a robot to learn affordances, it needs to observe the effects of its actions on the world. One approach is for the robot to explore the environment and ground its actions in its sensory space. This allows the robot to learn how its actions change the world around it and slowly build a library of models of its environment through interaction. Throughout this thesis, we utilize exploratory behaviors often seen in children and animals (Power, 2000; Lederman and Klatzky, 1993) to learn affordances. In particular, we use E.J. Gibson’s (2003) theory that learning affordances is “discovering distinctive features and invariant properties of things and events” by using basic primitive actions such as reaching or shaking to discover these properties.

While a robot should explore its surroundings to learn about its environment, blind exploration is inefficient and unrealistic in the real world where there exist hardware and

safety constraints. For example, if the robot were trying to learn the affordance open-able, even if the robot knew the exact location of the door handle, there are an infinite number of directions the robot can pull or push in a continuous action space to open the door. Aside from being physically intractable, even several hundred explorations per affordance would not be feasible due to the wear and tear of the hardware. Fortunately, in human environments, co-robots can leverage the people around them to provide guidance to their exploration much like how children can rely on their caretakers to provide guidance in their learning process (Vygotsky, 1978; McLeod, 2010; Wood et al., 1976; Mascolo, 2005). Vygotsky (1978) describes a “Zone of Proximal Development” where learning is “awakened... when the child is interacting with people in his environment”. Wood et al. (1976) perform a series of studies that build upon this notion and formally introduce the term “scaffolding” that consists of an “adult ‘controlling’ those elements of the task that are initially beyond the learner’s capacity, thus permitting him to concentrate upon and complete only those elements that are within his range of competence”. Both works conclude that scaffolding helps the learner to successfully and independently complete the original task. This thesis takes inspiration from these works to expedite robot affordance learning of manipulation tasks through the use of human guidance and environmental scaffolding.

1.2.3 Multisensory Input

Just as humans rely on multiple senses to interact with the world, robots should also use multiple sensory inputs to model the environment. In psychology literature, Lynott and Connell (2009) show that multiple sensory modalities are key to fully representing object properties. Work from Gaver (1993) shows that humans can perceive object properties through sound alone. Lederman and Klatzky (1993) prove the importance of touch and physical exploration when people are discovering properties of novel objects. Studies from Wilcox et al. (2007) show that a combination of visual and touch sensing improve object recognition while Ernst and Bulthof (2004) theorize that humans use a weighted combina-

tions of modalities (i.e. visual and tactile information) to determine object properties such as height.

While most prior work in affordance learning focuses on *visual* affordances, this thesis addresses multisensory models of affordances – what affordances *look*, *feel*, and *sound* like. We claim that by using multiple sensory modalities, the robot is more robust. Take, for example, a lamp as seen in Figure 3.8. While the robot could rely on visual information to determine if the light has been turned on, it can also utilize touch to detect the change in pressure, and sound to hear the click of the switch. This allows the robot to naturally develop contingency cases (e.g. if the light bulb had gone out). Furthermore, rather than have the robot blindly execute a trajectory, this allows a robot to adapt its control schemes to the environment by using feedback on each of its sensor modalities (e.g. pull until it feels a particular force, hears a click, or sees light). This thesis explores the importance of multisensory input for modeling affordances and the impact these modalities play in adapting affordance controllers for manipulation of everyday objects.

1.3 Thesis Overview

1.3.1 Thesis Statement

A robot can effectively manipulate objects in human environments by leveraging the structure of affordances and building adaptable controllers using multisensory input and human-guided exploration.

1.3.2 Contributions

To support this statement, this thesis makes the following contributions to the field of robot learning:

- **Human-Centered Framework for Robot Affordance Learning:** (Chapter 3) Throughout this thesis, we focus on how human teachers can guide robots throughout the

affordance modeling pipeline. We show in several user experiments that by utilizing human-guidance during the modeling process, robots can learn a wide range of multisensory affordances that typically have not been studied in robots (Chu et al., 2016a; Chu and Thomaz, 2017; Chu et al., 2017). Specifically, we demonstrate our human-centered framework (Human-Guided Affordance Learning) with naïve and expert human teachers in three different settings. These experiments result in teaching robots 5 different affordances across 11 different objects and actions.

- Human-Guided Robot Self-Exploration:** (Chapter 5) This thesis contributes several novel algorithms that enable robots to efficiently explore the environment using guidance from human teachers to learn affordance models for a diverse range of manipulation tasks. Specifically, in Chapter 5, we present experimental results with a robot learning 5 affordances on 4 objects using 1219 interactions. We compare three conditions: (1) learning through self-exploration, (2) learning from supervised examples provided by 10 naïve users, and (3) self-exploration seeded by the user input. Our results characterize the benefits of self and supervised affordance learning and show that a combined approach is the most efficient and successful (Chu et al., 2016b). Furthermore, we provide additional analysis of the variance seen across teachers during teaching. We provide a characterization of failure cases and insights for future work in learning from naïve end-users. (Chu and Thomaz, 2017)
- Multisensory Representation of Affordances:** (Chapter 4 and 6) Our contribution is developing a representation that integrates visual, haptic, and audio input to model affordances and quantifies the role each sensory modality plays in affordance modeling and control. Specifically, we present a system for learning haptic affordance models of complex manipulation skills. We model two specific affordances (openable and scoop-able) using five different actions over seven different objects using a force/torque (F/T) sensor mounted at the wrist of the robot. In Chapter 4, we show

we can successfully monitor a trajectory using haptic data to determine if the robot finds an affordance (Chu et al., 2016a). In (Chu et al., 2017) and in Chapter 7, we characterize the importance between visual, haptic, and audio data for affordance modeling and control. We show that modeling affordances using multimodal sensory information allows for more effective adaptation of manipulation skills than without.

- **Adaptive Object Manipulation using Multisensory Affordances:** (Chapter 7) The final contribution of this thesis brings together the different technical components and creates a control framework that uses a multisensory representation of affordances to enable a robot to adaptively manipulate objects of similar affordances (Chu et al., 2017). We show that we can break down affordances into subskill segments using keyframes. By modeling these subskills segments using left-to-right HMMs and multisensory inputs, a robot can adapt to 5 different drawer configurations and turn on 2 never-before-seen lamps.

CHAPTER 2

BACKGROUND AND RELATED WORK

To situate this thesis, we start by looking at robot affordance learning as a whole, then move into the specific areas of human-guided exploration, multisensory representation for affordances, and multisensory adaptable object control.

2.1 Robot Affordance Learning

Early research in robot affordance learning was controversial because of the many different ways to interpret “affordance”. The conflict stemmed from the evolving nature of the definition that was never formally set by J.J. Gibson (1977) in 1977. Sahin et al. (2007) tackle this confusion behind the definition of affordances and attempt to clarify and extend J.J. Gibson’s work specifically for robotics. Eventually, the paper advocates for the use of the ecological definition of affordance that most robotics researchers base their work on today by Chemero (2003). This framework relies on the effects that occur between agents, behaviors of the agents, and objects. To learn affordances, the most commonly accepted work comes from E.J. Gibson (2003), who claims that learning affordances is “discovering distinctive features and invariant properties of things and events” and that in children, basic primitive actions such as reaching or shaking, discovers these properties. This connects the field of affordance learning to another large field in robotics, developmental robotics, where researchers take inspiration from the psychology of child development and apply it to improving robots.

Early work in affordance learning for robotics took inspiration from developmental robotics and focused on using primitive actions to interact and learn about object effects. These established a framework for affordance learning using exploration. Fitzpatrick et al. (2003) used parametrized primitive actions to push or roll objects. Stoytchev (2005)

used a robotic arm to use tools to bring objects within reach. Dogar et al. (2007; 2008) tackled the traversability affordance using visual cues and wheel encoders. Montesano et al. (2008) and Lopes et al. (2007) learned affordances to be used for imitation. In an effort to use and plan with affordances, Kruger et al. (2011) developed a rich framework that allowed for affordances to be defined as low-level primitives as well as chained to perform high-level tasks. Hermans et al. (2013b; 2013a) investigated primitives for pushing objects on a flat surface. Moldovan et al. (2012) looked at the relationship between affordances for multi-object manipulation tasks. In the area of using scaffolding for affordance learning, Thomaz and Cakmak (2009) demonstrated the importance of scaffolding for learning affordances. Most of these early works in affordances focused on simple affordances and any system that used these learned affordances were mostly proof-of-concept experiments. While some of these works included proprioceptive information, the primary focus was on visual feedback and cues.

More recently, Koppula and Saxena (2013) used affordances to predict and anticipate human activities. However, the evaluation with the robot was proof-of-concept and the three main affordances were created using heatmaps of video images and external knowledge that linked locations in the environment to certain object affordances (e.g. drinkable, pourable, etc.) Katz et al. (2013b) demonstrated a system that could clear rubble with the same effectiveness of a human operator and significantly better than random and heuristic based pile removal algorithms. Furthermore, the system was tested extensively during experimentation. While Katz et al. used force compliant primitives, the primitives were hand tuned, specific to pile removal, and were not learned by the system. Furthermore, the forces were only used as thresholds and selected by the authors. Varadarajan and Vincze (2012) built on AfNet, an open affordance initiative, by providing semantic context and household manipulation objects, however, the work does describe how a robot could physically use the affordances. Ugur and Piater (2015) take a different approach and connect actions to symbolic planning. This allowed the robot to learn how to grasp and move objects,

which resulted in a robot that could plan to stack objects into towers. While the actions were learned, the actions were simple pick and place. In work from Wang et al. (2014), they looked at how to efficiently transfer affordances by reducing the amount of exploration needed to adapt affordance models. For a complete look at affordance learning with robots, there exists two general surveys where they focus on robot affordances in relation to psychology and neuroscience (Jamone et al., 2017), affordance research in developmental robotics (Min et al., 2016), and computation models of affordances in robotics (Zech et al., 2017).

The following sections go deeper into related approaches to tackling these challenges as well as how this thesis contributes to each area.

2.2 Learning from Demonstration

Before looking at the specific areas that this thesis contributes to, we take a brief look at the field of Learning from Demonstration (LfD). While this thesis does not make large contributions to the field of LfD, we heavily utilized LfD throughout this thesis. LfD is a field that focuses on how robots can learn from examples or demonstrations from teachers. This approach has become popular for motor skill learning (Pastor et al., 2011c; Kroemer et al., 2015; Akgun and Thomaz, 2015) and several surveys on LfD have been released (Argall et al., 2009; Chernova and Thomaz, 2014; Billard et al., 2008). In this thesis, we use a specific type of LfD algorithm that uses keyframes and kinesthetic teaching (Akgun et al., 2012a). This method of LfD allows a person to physically teach a robot how to perform skills (*e.g.* opening jars, pouring) by taking snapshots of the important points in a trajectory. This allows the robot to focus on the goals and subgoals of the skill rather than record the full demonstrated trajectory. Furthermore, it is important to note that, in this thesis, we utilize the skills generated *after* the execution and not the demonstrated trajectory to avoid capturing noise from the human teacher. As a result, the exact method of LfD can vary as long as the provided demonstration is a reasonable example that the robot can use to

execute a given skill.

2.3 Human-Guided Exploration for Affordance Learning

Many researchers are investigating how robots can explore the world. One relevant area of research is work on intrinsic motivation (IM) and curiosity-driven exploration. Early work (Oudeyer et al., 2007a; Vigorito and Barto, 2010; Schmidhuber, 1991) looked at using rewards and expectations to guide the exploration without any human supervision. Recently, Ugur and Piater (2017) have applied IM to learn the hierarchical structure of affordances such as various poking and stacking tasks. The most related work on IM from Ivaldi et al. (2014; 2012) and Nguyen and Oudeyer (2014), combined IM with human input. Methods using IM assume the existence of an easily-characterized reward signal, even though such reward signals can be difficult to define for hard-to-find affordances.

More recently, work in the field has focused on how to employ LfD and input from people to tackle these challenges (e.g. parental scaffolding (Ugur et al., 2015b) and crowd-sourcing (Krishna et al., 2016; Sung et al., 2015)). While crowd-sourcing uses teachers that are non-experts, these works focused on learning only visual information (Krishna et al., 2016) and information about the robot end-effector location (Sung et al., 2015) (as other sensory modalities are difficult to transfer through web interfaces). Nguyen and Kemp (2014) utilize human guidance by pointing to the location the robot should explore, however, the work requires an expert to provide specific actions for the robot to use during exploration. Ugur et al. (2015b) uses human guidance and self-exploration to modifying grasp actions using visual and haptic information. However, the actions were only acquired through an expert (the author). Ugur et al. (2015a) in later work follow up the study using naïve teachers, however, both works focused on simple affordances (e.g. push-able, lift-able for pick-and-place tasks). The published work in this thesis (Chu et al., 2016b; Chu et al., 2016a), demonstrates that human guidance from naïve teachers can be used to acquire a diverse set of actions (e.g. opening drawers, boxes, scooping) for human-guided

robot self-exploration with multiple sensory inputs.

2.4 Multisensory Representations of Affordances

Another area of research that has utilized the ecological and developmental robotic framework is in haptics. Haptics research often refers to influential work by Lederman and Klatzky (1993) where humans use “exploratory procedures” similar to primitive actions when interacting with novel objects. This provides a bridge between research in haptics and affordance learning that very few researchers have explored. There exists a large body of research that looks at visual learning of affordances (Krishna et al., 2016; Sung et al., 2015; Koppula and Saxena, 2013; Zhu et al., 2014) and this thesis uses much of the insight gained from these works to integrate visual information with audio and touch sensing. Since very little work has looked at characterizing what it means for a robot to learn these haptic and audio affordances, this section covers the most closely related works that use audio and haptic information for object classification.

The relevant work in haptic object recognition also uses exploration to interact with the environment to learn and classify haptic properties of objects (Torres-Jara et al., 2005; Sinapov et al., 2011a; Gemici and Saxena, 2014; Chu et al., 2013; Bhattacharjee et al., 2014; Fishel and Loeb, 2012). While some of these works use haptic information to obtain object properties (similar to object affordances), (Chu et al., 2013) and (Gemici and Saxena, 2014) do not directly learn how to use these properties once found and (Bhattacharjee et al., 2014; Bhattacharjee et al., 2013) only learned one affordance. To address these shortcomings, this thesis presents work in haptic affordances (Chu et al., 2016a) that learn different affordances with a wide variety of actions using only haptic information. Within audio affordance learning, the most relevant works (Sinapov et al., 2009; Torres-Jara et al., 2005; Romano et al., 2013) use exploration to determine different objects’ acoustic properties as well as understanding task completion (i.e. classifying if a stapler finished). However, these works do not look at trying to learn several affordances and instead only

try to use audio to classify when a specific task has completed.

Integration of all multisensory input (visual, haptic, and sound) in robot learning has not been an area that has been deeply studied, with most work focusing on how to integrate proprioceptive and visual information (Wieland et al., 2009; Li et al., 2013). The closest work to multisensory learning of affordances is work from Sinapov (2013; 2014b), Park et al. (2016), and Kappler et al. (2015). Sinapov (2013; 2014b) looks at object recognition and categorization using robot exploration and multisensory input. Park et al. (2016) look at how multisensory input can improve anomaly detection using HMMs. Kappler et al. (2015) explore how a bimanual robot can open bottles robustly by using multisensory information. While these works do not specifically look at affordance learning, they provide a good foundation for the work in this thesis.

In this thesis, we integrate multisensory information to learn affordances and show that a wide range of affordances can be modeled using only haptic information (Chu et al., 2016a). Furthermore, we provide a framework (Chapter 7) that uses multisensory data to perform complex manipulation tasks and show that the various modalities are key to modeling the effects of affordances.

2.5 Learning Multisensory Robot Controllers

Control from different sensor inputs has been studied for several decades, typically in the form of factory robots merging force and position feedback for manufacturing. However, these early methods of hybrid control (Mason, 1981; Raibert and Craig, 1981; Khatib, 1987), require an expert to hand-tune and specify each trajectory in very structured environments. To execute trajectories in *unstructured* environments, one popular approach is to use LfD to first teach the skill and play it back either exactly or after some exploration and modeling (Argall et al., 2009).

There are various ways to model a trajectory once it has been demonstrated by a human teacher. In the simplest instance, the proprioceptive information of the robot arm (i.e. joint

position and orientations) can be imitated and played back. To generalize the trajectory to a wider set of environments, additional information about the task is incorporated into the model. We can break down the methods to generalize a trajectory into three schools of thought: model-free methods, environment model based methods, and segmentation methods. We define model-free methods as methods that do not assume any information about the environment in which the trajectory is being executed and/or adapted to whereas model based methods use some inherent property of the task to model the trajectory (e.g. articulated joints). Additionally, each method is categorized by mode of input (*i.e.* visual, haptic, auditory).

2.5.1 Model-free Learning

Dynamic Motion Primitives (DMPs) have gained attention in the skill learning community. Introduced by Ijspeert et al. (2002) and Schaal et al. (2003), DMPs were used to learn attractor landscapes for imitation learning (*e.g.* teleop reaching movements). Several works sought to improve DMPs: online obstacle avoidance and goal adaption (Pastor et al., 2009a), adapting to Force/Torque (F/T) values on a robot wrist (Pastor et al., 2011a), refinement of DMPs using reinforcement learning (RL) (Kalakrishnan et al., 2011), learning and predicting successful DMP execution using interactive RL (Pastor et al., 2011d), expanding gain values to include multiple channels for compliance control (Kormushev et al., 2011a), adaption to bi-manual tasks (Gams et al., 2014), online error recovery using F/T sensing (Abu-Dakka et al., 2015), and initial work for stereotypical motion that stores sensory inputs (Pastor et al., 2012a). While some of these approaches merge haptic and visual information to learn a skill in various configurations, it is unclear if the approaches, which adapts a specific learned trajectory to a specific object, can generalize to vastly different objects with similar affordances.

Other methods for learning trajectories from human teachers include work from Calinon et al. (2007), where trajectories are modeled using Gaussian Mixture Models (GMMs)

and each Gaussian represents different segments of the skill. Playback is achieved using Gaussian Mixture Regression (GMR) over the GMM. Akgun and Thomaz (2015) use keyframes of demonstrations to learn a Hidden Markov Model (HMM) of the skill. Trajectories are generated by finding the shortest likelihood path through the HMM and splining the end-effector (EEF) poses. These works do not integrate any form of haptic information and it is unclear how these approaches can adapt to different objects of similar affordance type. On the flip-side, Rozo et al. (2013b; 2013a; 2011) model trajectories use only F/T sensing using HMMs and PHMMs, and GMRA (an adapted version of GMR that uses transitions learned from HMMs). Kruger et al. (2010) represents a set of actions that create the same effect using Parametric HMMs (PHMMs) and demonstrate that PHMMs can be manually strung together to form complex interactions. While this work does not use haptic information, it does embody a key point that this thesis explores: the ability to not only represent the action as a trajectory, but also as the effects the action causes. In work from Hangl et al. (2016), haptic information is used to learn complex skills by learning a hierarchy of preparation actions that connect to the effects of those actions.

2.5.2 Environmental Model-based Learning

This next section looks at representing skills from an environment model-based point of view where there is an underlying model (e.g. physics based) that is being learned from data. Katz et al. (2013a) uses visual information from an RGB-D camera while a robot interacts with objects to learn three possible kinematic models (prismatic, revolute, and disjoint). Pillai et al. (2014) learns kinematic models using RGB-D data from human demonstrations in clustered and unstructured environments. Martin-Martin and Brock (2014) implement a recursive hierarchical probabilistic method for articulation learning. Unlike the following sets of work, none of these approaches use haptic information. Sturm et al. (2010) used a force-torque sensor to learn a kinematic model of an articulated object (various doors). Jain and Kemp (2013) extend this work to create and use a database of

forces for different doors and used it across multiple agents (humans and two different robots). Wieland et al. (2009) merged the sensor spaces for opening doors and showed that using visual and haptic data worked better than using a single channel.

Sukhoy et al. (2012) learned the trajectory for sliding a card through a card reader with proprioceptive feedback. Sturm et al. (2010) learned a kinematic model for successfully opening various doors using a F/T sensor. While both only learned one primitive, they show that haptics could be used to improve actions directly through experience. Pastor et al. (2011c) learned and predicted the outcome of complex skills (e.g. flipping a box with chopsticks) by using pressure sensors and reinforcement learning on dynamic motion primitives. They extended (Pastor et al., 2011c) and introduced Associative Skill Memories (Pastor et al., 2012b) where they learned haptic feedback from demonstration of actions on objects. However, they did not learn and discover haptic affordances or properties of an object. Researchers have begun exploring how to model force and compliant dependent skills using LfD (Rozo et al., 2013c; Kormushev et al., 2011b). The focus, however, is on executing the specific skill and not on using trajectories to model the environment.

Researchers such as Ciocarlie et al. (2014), focused on directly using forces to adjust grasps to make them stable. Romano et al. (2011) used accelerometers to detect high frequency vibrations to improve grasping. Representing periodic motion (e.g. wiping, cranking a handle) can be seen as on the border of model-based trajectory learning as it assumes a periodic structure of the task but does not represent it in a specific model. These works include Petric et al. (2014) and Gams et al. (2015), where they teach a robot how to perform circular motions with a sponge to wipe a table. While the approach uses visual and haptic information to adapt a trajectory, it is unclear how this approach will transfer to different objects.

2.5.3 Segmentation Based Skill Learning

Segmentation based learning most directly relates to learning adaptable multisensory controllers. Segmentation algorithms split trajectories into basic action primitives and each segment is modeled using DMPs. Segments can be reused and represented in a hierarchy. Work in this area has focused on how to segment the provided trajectories (Niekum et al., 2015b; Niekum et al., 2015a), build representations of skills using these segments (Konidaris et al., 2012; Manschitz et al., 2015), and understand the transitions of skills based on the effects of each segment (Kroemer et al., 2014; Kroemer et al., 2015; Kappler et al., 2015; Su et al., 2016). Most of these works use some modified form of DMPs to model a segment and classifiers to predict transitions. Within this area of research, Righetti et al. (2014) and Chebotar et al. (2014) integrated F/T information with hand-coded segmented DMPs and (Kroemer et al., 2015; Kappler et al., 2015) included policy search and multimodal input to improve the DMPs of each trajectory segment.

Kroemer et al. (2015) and Kappler et al. (2015) use a multisensory approach to detect when and what skills to switch to and are most similar to contributions of this thesis. Kroemer et al. (2015) autonomously segments demonstrations using a modified autoregressive Hidden Markov Models (AR-HMMs) that represent skill segments by adding an additional edge between the hidden state and the previous observation. They introduce the state-based transitions autoregressive hidden Markov model (STARHMM), aimed to detect transitions based on the *effects* of the actions and used a combination of visual and haptic data streams. Kappler et al. (2015), fully expand the notion of Associated Movement Primitives (ASM) to include multisensory information about the effects of the actions (modeled as DMPs). The sensory inputs include visual, haptic, and auditory information. These data streams are used in conjunction with a hand-created manipulation graph to switch to the next ASM when a skill has failed. While these works generalize to different object configurations, it is unclear if these approaches will adapt to different objects. This thesis shows the importance of each sensory modality in adapting affordances in novel configurations.

CHAPTER 3

HUMAN-GUIDED AFFORDANCE LEARNING

In this chapter, we outline many of the challenges in robot learning that have not been addressed in prior work. We begin by looking at the overall challenges in robot affordance learning and then dive into the algorithmic solutions this thesis provides to address the subproblems of the field. We outline our human-centered approach to robot learning (Human-guided Affordance Learning), establish the affordance learning representation we use throughout this work, and introduce the machine learning algorithms used to model affordances.

3.1 Challenges in Robot Affordance Learning

There are several open questions and challenges in robot affordance learning that were briefly outlined in Chapter 2. This thesis aims to address several of these challenges:

- Most of the work in robot affordance learning has focused on visual information. This is largely due to two factors: (1) Gibson’s original definition describes affordances within the context of “visual perception”, and (2) affordances that rely on non-visual sensory channels are complex and require dexterous interactions with the environment. Works that include proprioceptive robot information (Fitzpatrick et al., 2003; Stoytchev, 2005) do so by including it in models either as part of a single state space with visual information or with hand-tuned thresholds on actions. The open question is “What role in the affordance representation do sensory inputs (visual, haptic, and auditory) play?”
- While several works in affordance learning have looked at large-scale learning of affordances, very few have performed a high-level task plan with a physical robot that does not require very carefully-tuned robot controllers. This disconnect between scale

and execution originates from the difficulty of getting robots to learn action controllers for each affordance. Prior work either focused on learning a small subset of simple affordances (e.g. push-able (Hermans et al., 2013b)) or building a large database from text datasets, images, and videos (Zhu et al., 2014). The challenge is how to ground a larger group of diverse affordances to allow robots to utilize the existing databases of affordances for task planning and execution.

We can break down these high-level challenges in robot affordance learning into specific subcomponents that relate directly to the contributions of this thesis:

- (1) Multi-sensory Representation:** *What role in the affordance representation do sensory inputs (visual, haptic, and auditory) play?* We explore this relationship first in Chapter 4, where we look at learning affordances using only haptic feedback and then more thoroughly in Chapter 7, where we do an in-depth comparison between touch, visual, and auditory input for modeling affordances.
- (2) Self-exploration:** *How can we quickly ground a large group of diverse affordances? How can this be done efficiently, with minimum effort from human teachers?* In Chapter 4, we introduce a method of guided exploration based on prior work (Thomaz and Cakmak, 2009) and extend it to a diverse set of objects and actions. We extend the method to include robot self-exploration in Chapter 5, where we introduce several novel human-guided robot self-exploration algorithms (Chu and Thomaz, 2017).
- (3) Non-experts:** *Can robots learn affordances from naïve human teachers as well as they can from expert teachers?* We explore this question more fully in Chapter 5, where 10 naïve end-users teach the robot several affordances. We explore the relationship between the quality of models learned and the differences between individual human teachers.
- (4) Adaptable Control:** Once a robot has grounded an affordance, it is unclear how a robot can then transfer this learned affordance without having to re-explore the envi-

ronment, unlike in prior work. In Chapter 7, we introduce RMAC, a control framework that uses multisensory data to model an affordance. RMAC addresses *how a robot can learn adaptable controllers that use multiple sensory inputs and demonstrations from human teachers*. We show that we can adapt and transfer the model to objects in various configurations as well as novel objects with similar affordances.

3.2 Affordance as a Learning Problem

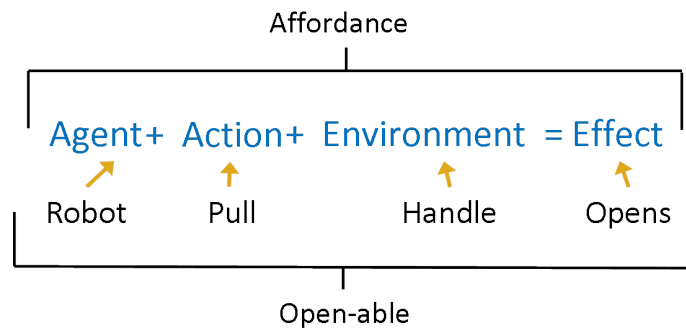


Figure 3.1: Visual representation of the components of an affordance in blue. Below it is an example affordance, open-able, broken down into the components of an affordance.

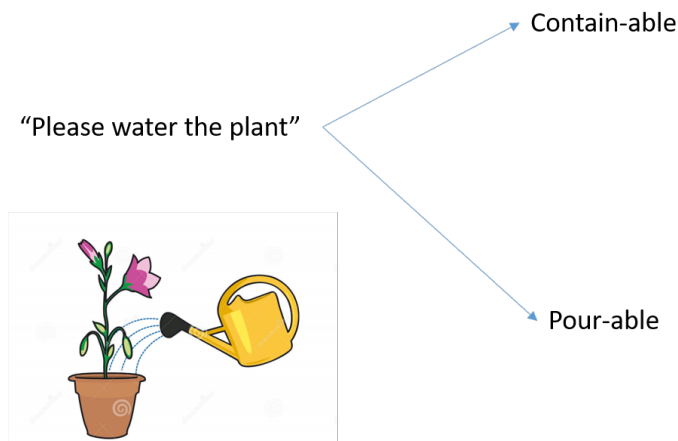


Figure 3.2: Affordances in relation to tasks and the importance of modeling each affordance to help achieve a task.

As described in Chapter 1, affordances are the “action possibilities” that appear between the agent and the environment. Affordances are a representational choice to model

skills as the relationship between effects and a set of actions performed by an agent on the environment. We can view this relationship in Figure 3.1, where an affordance is the combination of an **agent** performing an **action** on the **environment** to produce an **effect**. A specific example can also be seen in Figure 3.1, where the open-able affordance consists of a robot (agent) pulling (action) on a door handle (environment) to open the door (effect). Affordances enable us to communicate object properties and tasks to the robot (e.g. shown in Figure 3.2 where, given a task to water a plant, a robot can reason that it needs an object with a contain-able and pour-able affordance). Throughout the rest of the thesis, we define a skill as a low-level trajectory that achieves a specific goal and we use skill and action interchangeably. We define a task as a high-level goal that requires the execution of several skills.

For a robot to learn affordances, it needs to interact with the environment and observe the effects of that interaction. This interaction needs to be done by the robot (as opposed to the robot merely observing a human performing the skill) because affordances are action possibilities that occur between the environment and the agent. More concretely, there exist many objects that have affordances for a human that do not exist for all robots (e.g. jar lid is too wide for the robot to grasp). Once a robot interacts with an object and observes the effects of that interaction, the agent can learn what the environment affords. In our work, a robot (agent) performs a set of actions $A = \{a_1, \dots, a_N\}$ on a set of objects $O = \{o_1, \dots, o_M\}$ to model the effects that a_i can have on o_j , where $i = \{1, \dots, N\}$, $j = \{1, \dots, M\}$, and N and M are the number of actions and objects respectively. We assume the effect of an object-action (o_j, a_i) pair is labeled as a positive or negative example of the affordance. Thus, it is a supervised learning problem and the resulting model can recognize the successful (or unsuccessful) interactions of an object-action pair. This representation can be seen in Figure 3.3.

The overall goals of affordance learning in robotics fall into several categories: (1) grounding the effects of actions for specific affordances (e.g. the visual feature change

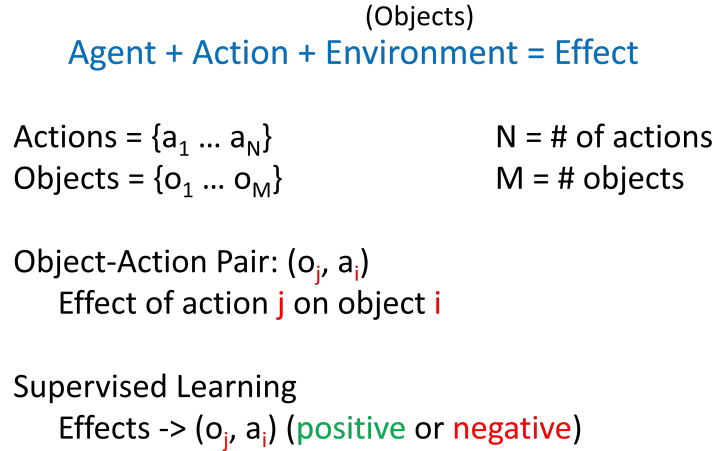


Figure 3.3: Affordances can be broken down into learning object-action pairs. This results in a supervised learning problem.

when an object is push-able) (2) using these learned models to test an object for that affordance (e.g. can I use this stick to stir?) and (3) giving the robot a task and having the robot find the objects with the suitable affordances to accomplish it.

3.3 Human-Centered Approach

To address these challenges, we take a human-centered approach that provides several direct benefits for robot affordance learning. Affordances allow people to connect intentions and goals to the actions that the robot needs to perform to accomplish these goals. This provides a natural communication channel between people and robots because people are goal-oriented (Csibra, 2003; Meltzoff and Decety, 2003) and can easily demonstrate to the robot the tasks they want the robot to accomplish. More importantly, aside from making communication easier for the human teacher, the robot benefits from this interaction by utilizing the information given during the demonstration. This allows us to go further than prior work in learning simple affordances (e.g. push-able and tip-able). In particular, this thesis explores a diverse set of complex affordances (*e.g.* open-able, turn-on-able, scoop-able), which require varied and difficult actions that are necessary to successfully complete real-world tasks.

For example, for a robot to ground how to turn on the lamp seen in Figure 3.8 through exploration without any outside knowledge, would require the robot to exhaustively explore the object space. This exploration space is immense and, even if we were to limit the range of exploration (e.g. just pulling on the chain), there exists an infinite number of directions and distances to pull in a continuous action space. Furthermore, an expertly hand-tuned controller is required to acquire the action to explore the lamp. This limitation has restricted many prior works in learning complex affordances. However, by having a human teacher demonstrate how to turn on a lamp, the robot now has information on (1) where it should explore (e.g. downward direction), (2) what effect it should be looking for (e.g. force pressure change, light change, etc.), and (3) the action to perform (e.g. grasp the chain and pull). In the rest of this thesis, we describe three separate methods that use this human-centered approach to contribute to the field of robot affordance learning.

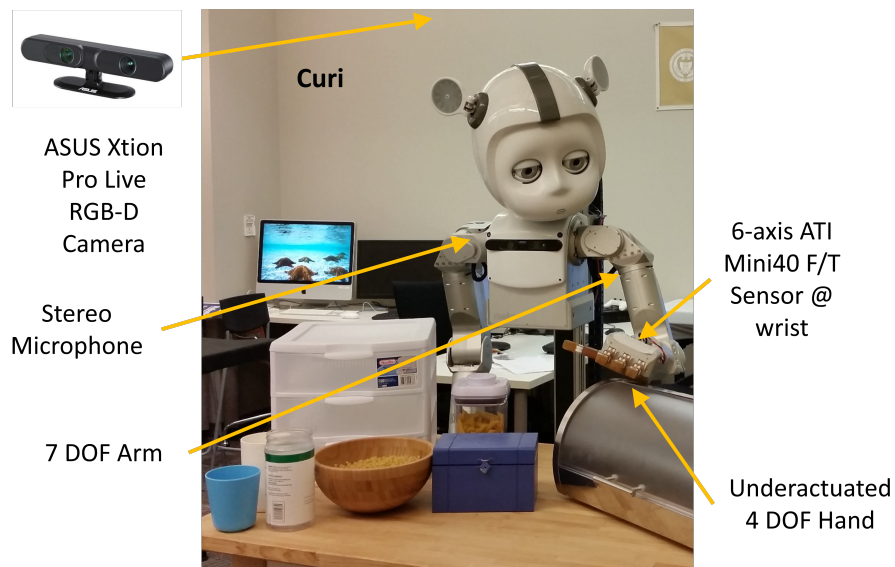


Figure 3.4: “Curi” robot used in several experiments. The various sensors used throughout the experiments include F/T sensing, RGB-D data, and audio.

3.4 Hardware

For the experiments through this thesis, we use two different robots, “Curi” and “Prentice”. They are each described in this section.

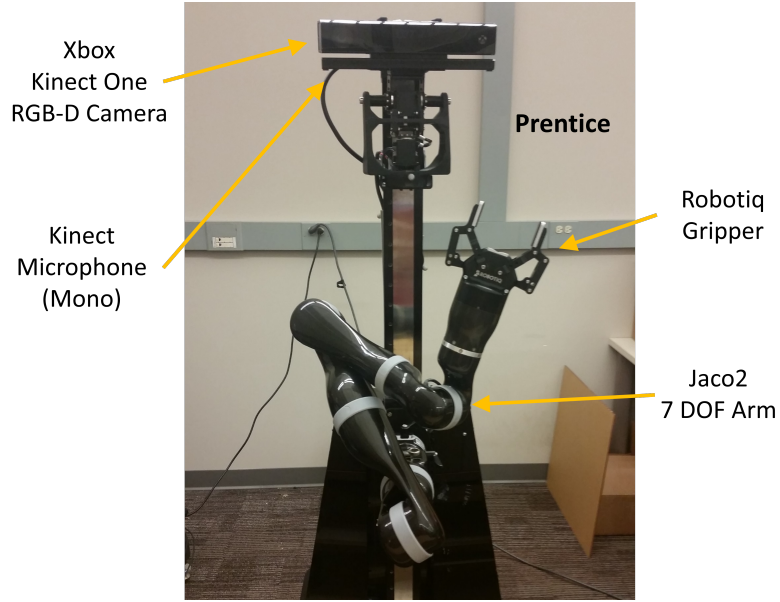


Figure 3.5: “Prentice” robot used in several experiments. The various sensors used throughout the experiments include RGB-D data, gripper position, and audio.

3.4.1 “Curi” Robot

For our experiments, we used the robot “Curi”, seen in Figure 3.4. Curi has two 7 degree-of-freedom (DOF) arms, each with an under-actuated 4 DOF hand. The arm can be controlled by physically moving it in a gravity-compensated mode and used to kinesthetically teach the robot actions. An ATI Mini40 Force/Torque (F/T) sensor is mounted at each wrist, an ASUS Xtion Pro RGB-D sensor is mounted above the workspace, and microphones are mounted in its chest.

3.4.2 “Prentice” Robot

We also use the robot platform, “Prentice”. Prentice has one Kinova Jaco2 7 DOF arm with a Robotiq pinch gripper. The arm can be controlled by physically moving it in a gravity-compensated mode. Prentice has an XBox Kinect One RGB-D sensor mounted to a pan/tilt unit. To record the different modalities for the experiment, we use the Jaco2 internal forward computed kinematics to record the gravity compensated wrench at the wrist, the Kinect2 to record visual and audio data, and the gripper position from the Robotiq gripper.

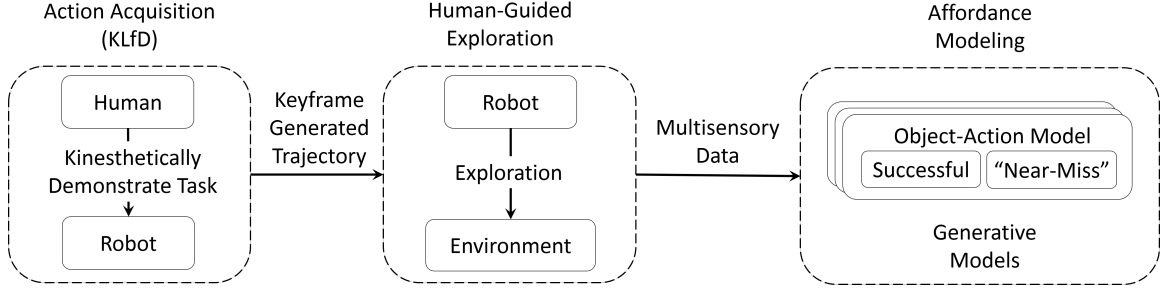


Figure 3.6: High-level pipeline of our human-centered approach to learning affordances. We use KLfD to acquire actions from humans end-users. These demonstrations are used by the robot to explore the environment and collect multisensory data. This data of both successful and “near-miss” exploration is then used to model object-action pairs.

3.5 High-Level Pipeline

Throughout this thesis, we take a common approach to obtain actions, gather data, and model the environment. This section walks through the framework shown visually in Figure 3.6 and outlines the methods that are each discussed in greater detail in the subsequent chapters. We can also see how the high-level approach relates to the contributions of robot learning in Figure 3.7.

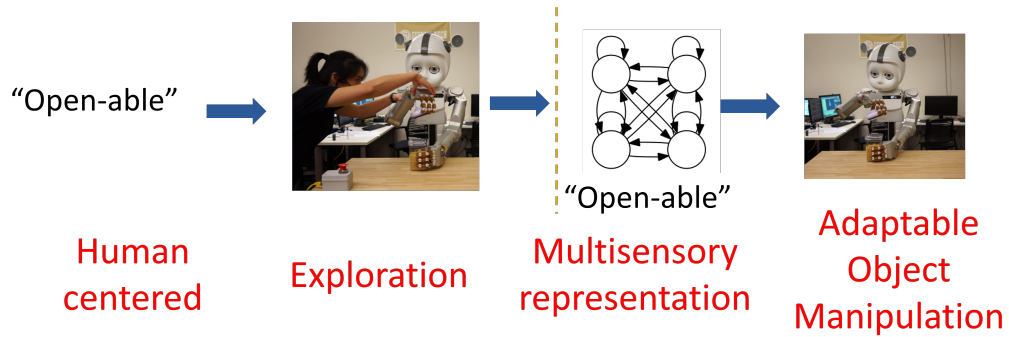
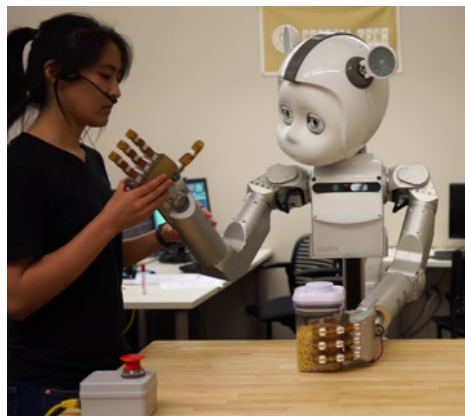
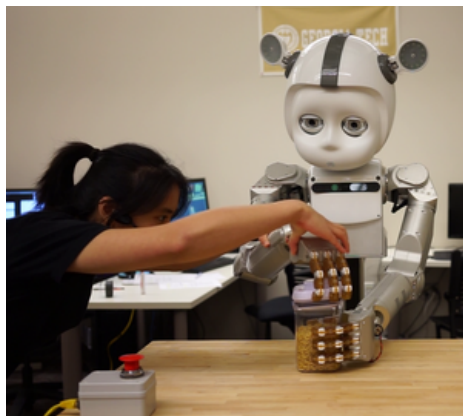


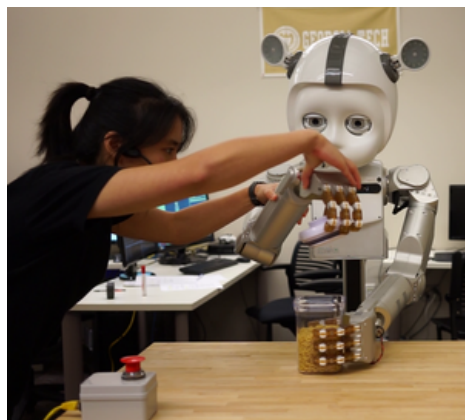
Figure 3.7: High-level view of contributions and overall approach to robot learning. The robot is given demonstrations from non-experts and uses these demonstrations to explore the environment and collect data from various sensors. This multisensory data is used to model the environment and the models are used to perform manipulation tasks on new objects.



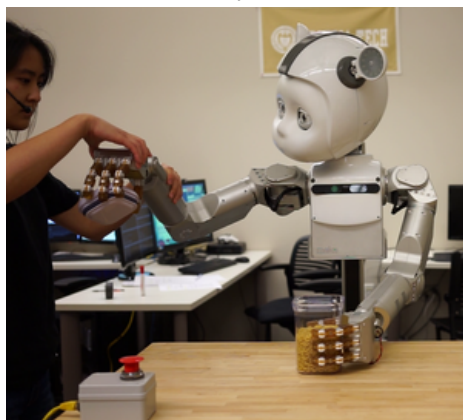
(a) “Start here”



(b) “Close your hand”



(c) “Go here”



(d) “End here”

Figure 3.8: Kinesthetic keyframed-based LfD for the open-able affordance on the pasta jar. Keyframes are recorded using verbal commands listed below each image.

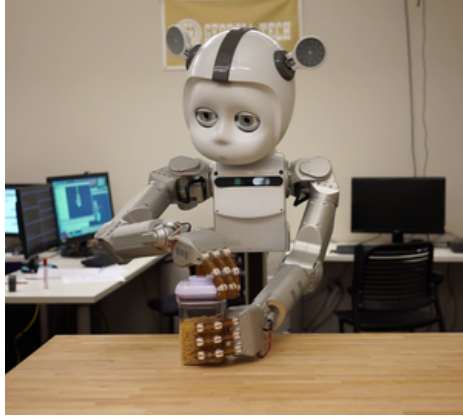
3.5.1 Action Generation

In our approach, we use LfD to acquire actions for the robot to use during exploration. Specifically, a teacher provides a demonstration of an action by physically guiding the robot to perform it (as opposed to observations of a human performing the action). This highlights a key point that affordances are “action possibilities” that occur between the agent and the environment. There are likely many objects that have affordances for a person that our robot would be unable to achieve (e.g. jar is closed too tightly for the robot to open). Particularly for multisensory affordances, it is essential that the robot successfully explore the environment to learn what the effects of particular object-action pair *feel, sound, and look like to the robot*. LfD for affordance learning is one novelty of our work and allows us to quickly program several primitive actions. This method is used throughout all methods in this thesis and gives us the ability to experiment with affordances that have typically not been studied.

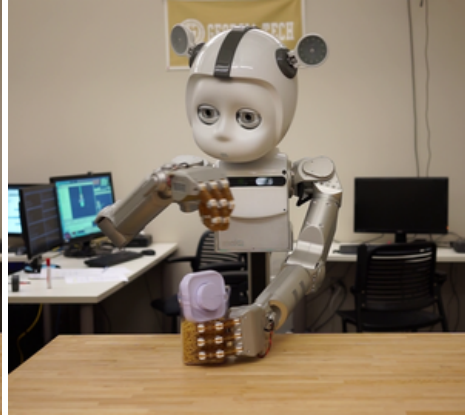
We use a keyframe-based LfD approach (Akgun et al., 2012a), whereby a teacher demonstrates each action by physically guiding the robot and marking salient points of the action. This process can be seen visually in Figure 3.8. During these points, snapshots of the joint states are stored as keyframes (KFs). To replay a demonstration, the KFs are splined together into a single trajectory using a quintic spline at an average velocity of 0.1 radians/second. The velocity was pre-selected to execute smoothly on the robot and applies for all actions. The robot executes the trajectory autonomously on the object during playback (Figure 3.9). This guarantees no external noise (e.g. forces) are felt during data recording.

3.5.2 Data Collection: Robot Exploration

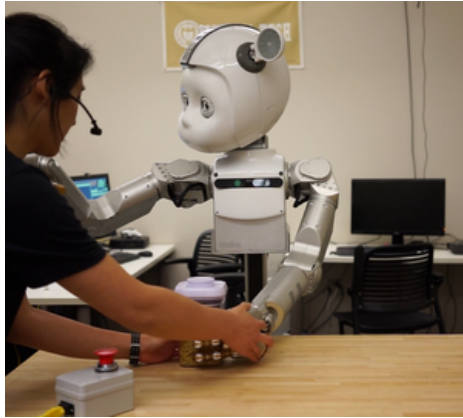
To collect the data to perform supervised learning for the object-action pairs, the robot needs to use the actions it learned. The data must contain diverse examples of the object-action pair. Throughout this thesis we introduce several different methods of human-guided



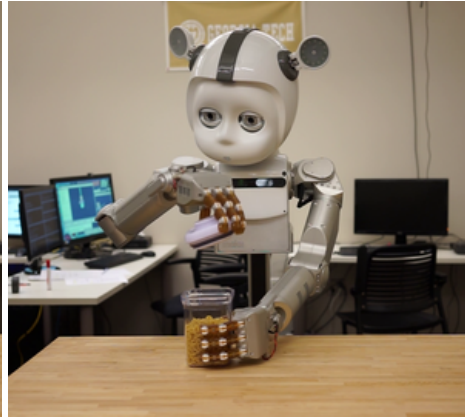
(a) Approach lid



(b) "Near-miss"



(c) Change environment



(d) Success

Figure 3.9: Curi executing demonstrated trajectory on the pasta jar. (b) Curi misses the first time, (c) a person adjusts the object, and (d) Curi succeeds

exploration. In similar prior work (Thomaz and Cakmak, 2009), human-guided exploration yielded high-quality learning examples that provided focus for exploration within a very large search space. In this thesis, we build on prior work and show that human guidance during this data collection step is crucial to reducing the search space of complex object-action pairs and can also be used to efficiently and effectively guide robot self-exploration.

Specifically, in Chapter 4 and Chapter 7, we use environmental scaffolding where the robot repeats the demonstrated action several times. For each interaction the human moves the object to perturb the action context slightly (Figure 3.9). This is a teaching interaction known as environmental scaffolding (Mascolo, 2005). In Chapter 5, we introduce several novel algorithms that decrease the burden from the human teacher and include robot self-exploration (Oudeyer et al., 2007b) in this process.

3.5.3 Affordance Modeling

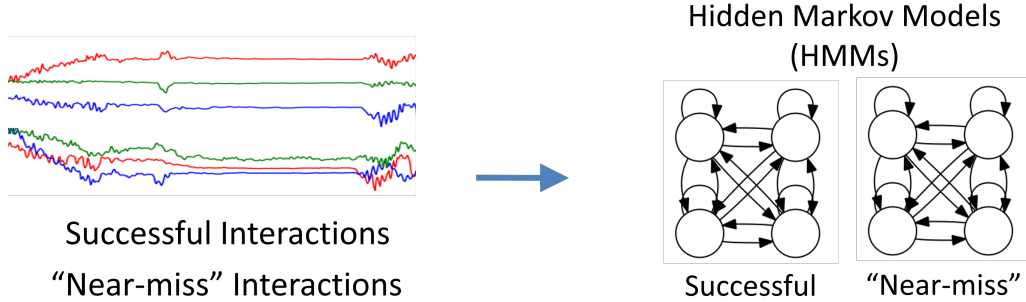


Figure 3.10: Object-action pairs represented using Hidden Markov Models. We learn two HMMs per object-action pair to represent success and “near-miss” interactions. The effect space depends on the modality (haptic, visual, audio) for each experiment.

Each interaction during the exploration phase generates a continuous signal from various sensors mounted above or on the robot (*e.g.* microphone, cameras, etc.). We model these sensory readings as the effect of the object-action pair by using Hidden Markov Models (HMMs) (Rabiner and Juang, 1986). We build two HMM models (Figure 3.10) for a given object-action pair: one HMM from examples of successful interactions and another HMM from examples where the object-action execution failed to find the affordance. In

this thesis, we assume unsuccessful interactions are informative “near-misses” of the action. This model thus characterizes what the object-action pair effects look like when the robot does not find the desired effect. Importantly, this is not a model of all failures, which would be a huge class, but of the much smaller and likely more informative class of boundary case failures that are close in action space to success (Grollman and Billard, 2012; Grollman and Billard, 2011) (e.g. a lid slipping from the hand when lifting). Furthermore, modeling near-misses can provide knowledge to detect when a trajectory begins to deviate from success, enabling a robot to adapt in real-time. A benefit of modeling both success and “near-miss” is that the decision per object-action pair can be made by comparing the log-likelihood of these two models. Given that we have multiple object-action pairs per affordance, the robot can use any/all combinations of object-action pairs previously learned (e.g. comparing the log-likelihood from all HMMs) to determine if an object has an affordance.

We use HMMs to model the sensory information throughout this thesis because of the time-varying nature of the data and the ability to generate trajectories of an action by sampling from the HMMs. We use various HMMs, which include ergodic and left-right right HMMs. The parameters of an n -state HMM, (A, B, π) , are estimated using Expectation Maximization (EM) where A is the transition probability distribution ($n \times n$), B the emission probability distributions ($n \times 1$), and π the initial state probability vector ($n \times 1$). We model the emission probability distribution using a continuous multivariate Gaussian distribution. Each chapter will go into detail on the specific observation space (O) used for each method.

Additionally, HMMs are a good representational choice for our defined object-action pair representation of affordances. Specifically, we choose to represent affordances as effects that are the result of an action on the object. If we model the effects of an action using an HMM, the hidden states, Z_n , represent where along the trajectory (action) the robot is currently executing relative to the object. The observation space represents the effects that

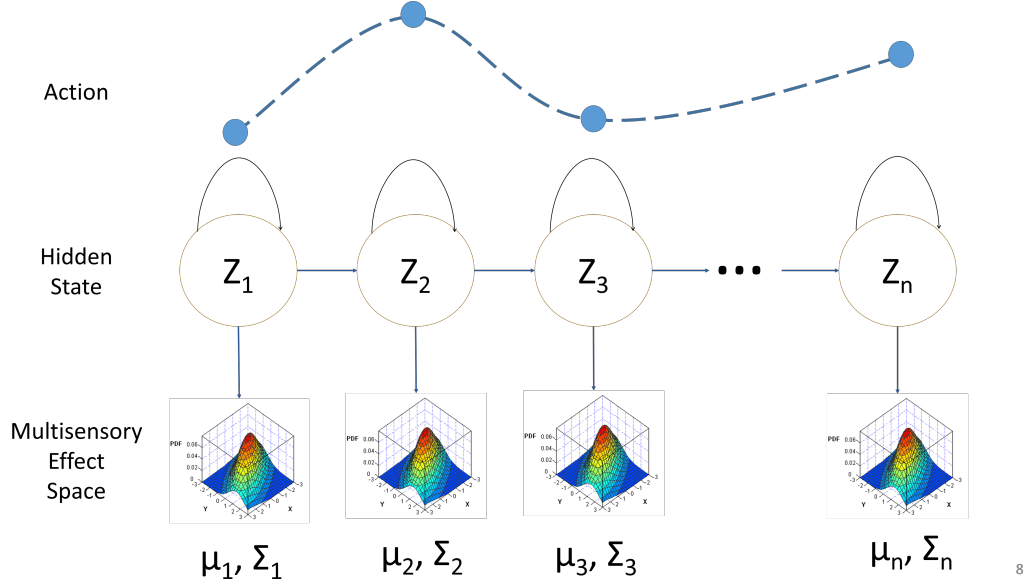


Figure 3.11: Visually representation for Hidden Markov Models as affordances. Where along the trajectory (action space) can be estimated as the hidden state, Z_n , which is represented using multivariate Gaussians of the multisensory robot data of the effects of the action.

the robot is currently experiencing, which are emitted based upon the robot’s pose in the action space. By modeling the emission probability distribution as a continuous multivariate Gaussian distribution, we can capture the importance of each sensory input over time. This relationship can be seen visually in Figure 3.11.

3.5.4 Evaluation

To evaluate affordance learnings, there are several different aspects of affordances that need to be taken into account. The goal of learning affordances is to give a robot a library of object-action pairs it can use to perform tasks. Evaluating whether the learned object-action pairs have achieved this goal requires us to break down the various ways in which a robot can use an affordance model. We can look at a concrete example introduced in Section 3.2 where the robot is asked to perform the task of “watering a plant”. To achieve this task, a robot must find an object with the two affordances: contain-able and pour-able. To determine if an object has these affordances, the robot must test objects and

Table 3.1: Metric Equations

Precision	Recall	F_1
$\frac{tp}{tp+fp}$	$\frac{tp}{tp+fn}$	$2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

compare it to that affordance’s existing object-action pair. Specifically, the robot must determine if performing the action of pour, with a given object, creates the effect that has been previously learned. There are two distinct situations the robot can encounter when testing an object for an affordance: (1) the robot finds the same object it had used previously when learning the object-action pair (2) the robot encounters a new object and must determine whether the object has the desired affordance.

Monitoring

In situation (1) where the robot finds the same object, the evaluation is relatively straightforward. To test the object, we determine whether the previously learned object-action pair has sufficiently captured the effects between the robot’s action and the object. This becomes a standard binary classification task where we label never-before-seen successful and unsuccessful interactions with the object. In this situation, we use the standard binary classification metrics of precision, recall, and F_1 as defined in Table 3.1 where tp is the number of true positives, fp false positives, tn true negatives, and fn false negatives. We also include overall accuracy. Precision is a measure of quality (e.g. how accurate is the model when it does label an interaction with the drawer as open-able?) and recall is a measure of completeness (e.g. of all interactions with the drawer, did the model miss any instances of open-able?). It is important to note that while this evaluation is used to determine how well we modeled the object-action pair, it can also be used during task execution. In particular, we can use binary classification to monitor whether the robot has succeeded in performing the skill (e.g. pouring).

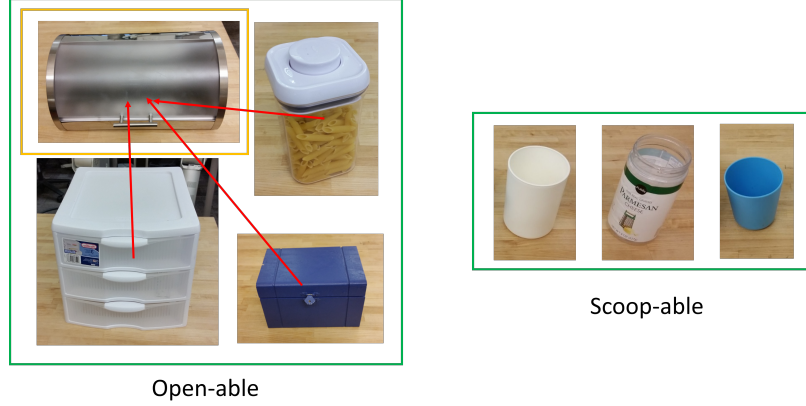


Figure 3.12: Offline transfer evaluation where the open-able object-action pairs from the jar, box, and drawer are tested on the breadbox. Similarly, this can be done with the different cups for scoop-able.

Transfer to Novel Objects

In situation (2), the robot encounters an object it has not seen before and must determine if this novel object has the desired affordance. There are several challenges that exist with this situation. While the effect of an object-action pair is common across objects with the same affordance (*e.g.* all lamps emit light), the action associated with each object may differ. For example, the affordance open-able may be associated with different actions depending on whether you’re trying to open a jar, a drawer, or a box. To test a new object for an affordance, the robot must understand how to adapt its existing action to the new object. However, adapting existing trajectories to new objects is an active research field on its own (Taylor and Stone, 2009; Pastor et al., 2009b; Pastor et al., 2011b). In this thesis, we address this challenge through methods described in Chapter 4 and Chapter 7.

In Chapter 4, we introduce an evaluation that simulates transfer across objects without adapting the robot’s actions. This is done by comparing one object’s existing object-action pairs to that of another object with the same affordance. Specifically, we test whether the object-action model of opening a drawer with a pulling action can be used to classify the sensory stream of opening a breadbox with a lifting action. Both object-action models fall under the affordance of open-able and this evaluation gives insight into whether the effects

of affordances can be transferred across objects (*i.e.* can the robot determine if the effects of open-able are universal across objects). This can also be seen visually in Figure 3.12 where we evaluate the open-able object-action pairs from the jar, box, and breadbox drawer. Similarly, this can be done with the different cups for scoop-able.

While this offline affordance testing can give insight into affordance transfer, the robot cannot use the learned object-action model on a novel object to complete a task. To do so, the robot must generate actions from its existing model and adapt the action to the new object. To perform this transfer, we introduce an algorithm in Chapter 7 that breaks down object-action models into subskill components that allow the robot to generate motion. In this situation, the evaluation metric is no longer binary classification but, instead, execution success of an existing learned object-action model on novel objects.

The following chapters dive into the different algorithmic contributions to robot learning using multisensory input and human-guidance.

CHAPTER 4

LEARNING HAPTIC AFFORDANCES FROM DEMONSTRATION AND HUMAN-GUIDED EXPLORATION

To address the question of “What role in the affordance representation do sensory inputs (visual, haptic, and auditory) play?”, this chapter looks at learning haptic models of affordances – what successful and unsuccessful interactions of a object-action pair *feel like*. The robot perceives these affordances using F/T sensors. The haptic model of an object-action pair is complementary to visual affordances. While both require acting on the object to learn an affordance, a learned visual affordance can be used to select “action possibilities” *prior* to interacting with the object; whereas haptic models can only provide information on possibilities *during* the interaction. Furthermore, some affordances are visually difficult to detect, but are salient through force sensing. For example, one can often predict whether a door opens by pulling vs. pushing by looking at the handle. However, there are many cases where visually this is not possible to detect and only through the forces felt when interacting with the door can this be determined. Without this information, a robot could easily damage itself or the door. However, together, visual and haptic information can provide a richer set of possibilities for the robot to find and utilize.

This chapter also looks at how to answer the question of “How can we quickly ground a large group of diverse affordances?” In this chapter, we explore the approach outlined in Chapter 3 that leverages a human teacher to assist the robot in rapidly exploring a variety of objects to learn haptic affordances. This chapter contributes a system that (1) uses a human teacher to both rapidly acquire actions *and* explore objects for learning affordances and (2) uses *only* haptic sensing to identify multiple affordances on unseen objects. We use the system to perform 5 different actions over 7 different objects to build object-action models for the haptic affordances of “open-able” and “scoop-able”. We show that the learned object-



Figure 4.1: Our experimental platform, Curi, with the various objects it learns affordances for in this chapter.

action models achieve good cross-validation performance with 4 of the 7 object-action pairs achieving a perfect F_1 score. Also, by leveraging the set of object-action models per affordance, we perform leave-one-object-out testing to identify affordances with an average accuracy of 67% for scoop-able and 53% for open-able, with haptic sensing alone.

4.1 Approach: Learning Haptic Affordances

To learn haptic affordances, a robot must successfully interact with objects and build a model of the interaction. Our approach has four components (Figure 4.2). We use the pipeline described in Chapter 3 and add additional detail specific to this chapter. Specifically, we use environment scaffolding to help the robot gather data. We also include the testing framework for offline affordance monitoring described in Section 3.5.4.

(1) Action Acquisition: We use LfD to quickly show the robot an exploration action to perform (Figure 3.8). Most prior work use one or two simple primitives (e.g. push is popular), whereas here we have five primitive actions with a range of complexity.

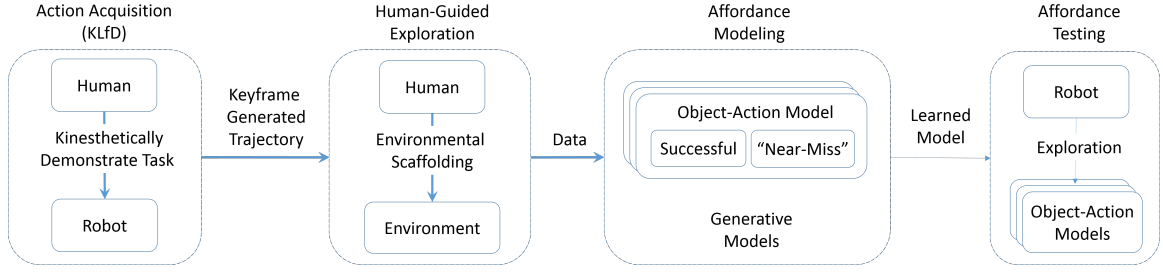


Figure 4.2: Components and information flow of the system. Action Acquisition builds an action trajectory from human demonstration. In Human-Guided Exploration, environmental scaffolding yields successful and “near-miss” interactions. This data is then used during Affordance Modeling to build a set of generative object-action models where in Affordance Testing, they are used to determine if an object has an affordance.

(2) Human-Guided Exploration: As described in Chapter 3, we use human-guidance to perform environmental scaffolding (Mascolo, 2005) to allow the robot to quickly gather interactions with the environment. Aside from quickly collecting high-quality data from the interaction, this step provides an important piece of information to the robot. With no initial information about what affordance the robot is looking for, guidance inherently provides the specific affordance of the action-object pair; it defines the objective for the robot to achieve.

(3) Affordance Modeling: We generate two HMMs (success and “near-miss”) as described in Chapter 3. In this chapter, we focus on learning affordances that have a diverse range of actions that can achieve the affordance. By modeling each object-action pair as two HMMs, this allows the robot to not only learn **what** an affordance feels like, but also provides a library of actions for **different ways** in which the affordance has been achieved. For instance, there are multiple ways for an object to be open-able (e.g. open a drawer by pulling, open a jar by twisting the top) and by modeling each of these methods, the robot now has access to a library of actions to explore a new object to find the affordance.

(4) Affordance Testing: As described in Section 3.5 we evaluate the learned models by monitoring the effect of the execution of each object-action pair for that affordance. We do this within object-action pairs as well as across object-action pairs.

The remainder of the chapter, is focused on our implementation and validation of each

of the four main components mentioned above.

4.2 Hardware Platform

We use the robot platform “Curi” and in this chapter, utilize just the left arm. Curi is positioned in front of a table with an object on it (Figure 4.1). Above the table is a ASUS Xtion Pro RGB-D sensor. We segment the objects on the table using the point cloud data. The object’s pose (position and orientation) and bounding box are recorded.¹

4.3 Action Acquisition

We taught Curi two affordances (**scoop-able** and **open-able**), which results in five separate actions to interact with seven different objects. The actions range in complexity starting with simple actions that can be easily executed by the robot (e.g. pushing) to more realistic actions on objects that can be found in homes (e.g. scooping pasta, opening drawers). The first action is on the objects shown in Figure 4.3, where the same scooping motion is repeated using three different, but similar objects. We then increase the difficulty of the task by selecting an affordance that require four different actions over varying objects. The set of objects shown in Figure 4.4 are all open-able, but require different low-level actions. The object-action pairs are listed in Table 4.1. To increase the stability of some of the lighter objects during interaction, Curi’s non-functional right arm was propped up and the weight of the arm prevented the objects from sliding. This was done for the objects Pasta Jar, Wooden Box, and the bowl of macaroni during all of the scoop actions.

4.4 Human-Guided Exploration

Next we collect a dataset of haptic information during object-action interactions for each of the seven object-action pairs described in Table 4.1. Each action was executed by the

¹While the visual data is not used in this chapter, we record it to allow for future integration with systems using visual affordance learning.

Table 4.1: Affordances

Object	Action	Effect	Affordance
Cup 1	Scoop	Macaroni in container	scoop-able
Cup 2	Scoop	Macaroni in container	scoop-able
Parmesan Bottle	Scoop	Macaroni in container	scoop-able
Pasta Jar	Lift	Cap Lifts	open-able
Drawer	Pull	Drawer slides	open-able
Wooden Box	Push1	Lid opens	open-able
Bread Box	Push2	Lid opens	open-able



Figure 4.3: **Scoop-able objects:** Left-to-right - Bowl of macaroni, Cup 1, Cup 2, Parmesan Bottle

robot 20 times on the same object, such that 10 interactions successfully found the affordance on the object and 10 were unsuccessful “near-misses”. This was done by moving the object around by the human² and hand labeling when the interaction did or did not find the affordance. As “near-misses” occur naturally when executing a skill, overall, extra interactions were not necessary to achieve an even split of successful and “near-miss” examples. For the open-able affordance, “near-misses” often included interactions where Curi missed the handle or lid of the object. For the object-action pairs in the scoop-able affordance, “near-misses” were instances where the cup dragged along the macaroni, but did not get any macaroni in the cup. During each interaction, the robot records: F/T data from the wrist sensors (example in Figure 4.5), object pose information, and all joint positions.

As mentioned previously, in some cases, it is visually very difficult to see when these actions find the affordance and could easily fall within the noise of a RGB-D sensor. For example, it was difficult for even the experimenter (the author) to detect the change in

²in this chapter, the teacher is the author



Figure 4.4: **Open-able objects:** Left-to-right - Pasta Jar, Drawer, Wooden Box, Bread Box amount of macaroni in the large bowl and in the cup. This suggests that using haptic feedback during action execution is important to fully understand the objects and object affordances.

4.5 Learning Haptic Affordance Models

With the data collected, we build two haptic models (success and “near-miss”) for each of the seven object-action pairs. This results in two different haptic affordance models: one for open-able and one for scoop-able.

By training two models for each skill, we can use the HMMs to monitor the success or failure of a skill. We can use the success of the skill to determine if an object has an affordance that the robot can perform. To train HMMs, we split the data into train and test sets. The detail and results of the training are described in this section.

4.5.1 Hidden Markov Models

As described in Chapter 3, we represent each object-action object pair using two HMMs. In this chapter, the observation state-space O is $[F_x, F_y, F_z, T_x, T_y, T_z]$ where F are the forces and T the torques. For our implementation, we used the Python library scikit-learn (Pedregosa et al., 2011).

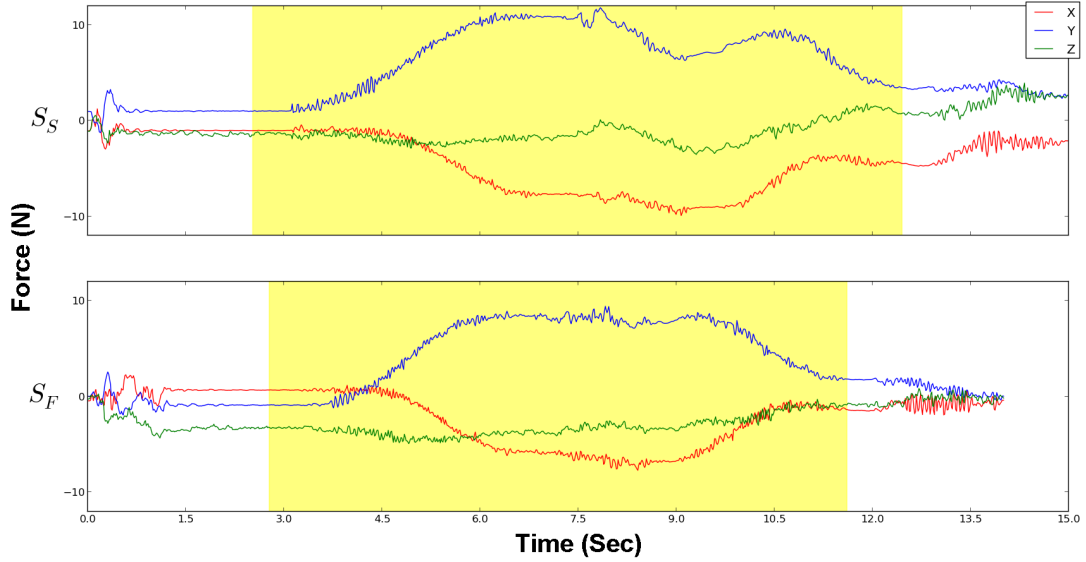


Figure 4.5: An example of force data we collect from the sensor during a scooping skill. The yellow shaded portion indicates the time period when the hand is in contact with the object. The top graph is a success and the bottom a “near-miss”.

4.5.2 Training

We split the data randomly into a train (80%) and test (20%) set for each object-action pair and each type of model (i.e. 8 training and 2 testing interactions for both success and “near-miss”). We select the optimal number of states (between 2-6 states inclusive) for the HMMs by performing leave two-out cross-validation (CV). With 8 interactions in the training set, this results in 28 CV sets where each set has a different variation of 2 trajectories removed for testing.

4.5.3 Modeling Results

For each object-action pair, we look at whether the models can determine success versus “near-miss” for each test interaction. Per Section 3.5, correctly monitoring the success and “near-miss” of an object-action pair allows us to test for affordances in objects. Therefore, to evaluate our models, we look at how the models perform at monitoring test interactions. We use the standard binary classification metrics of precision, recall, and F_1 as defined in Table 3.1 in Chapter 3. The interaction is classified as successful if the log-likelihood of the

Table 4.2: Affordance Skill Monitoring Results

Object-Action Pair	Precision	Recall	F_1	Accuracy
Cup 1-Scoop	0.50	1.00	0.67	0.50
Cup 2-Scoop	1.00	1.00	1.00	1.00
Parmesan-Scoop	0.67	1.00	0.80	0.75
Pasta Jar-Lift	1.00	1.00	1.00	1.00
Drawer-Pull	1.00	1.00	1.00	1.00
Wooden Box-Push1	1.00	1.00	1.00	1.00
Bread Box-Push2	0.00	0.00	0.00	0.50

successful model is greater than the log-likelihood of the unsuccessful “near-miss” model.

The resulting scores for correctly determining each object-action pair can be found in Table 4.2. Overall, each of the models perform well at determining if the test interaction within each pair was successful versus “near-miss”, with four object-action pairs achieving a perfect F_1 score and another with a score of 0.75. This could be attributed to the fact that “near-misses” and successes have very unique F/T readings. For example, when lifting the lid off of the jar, the robot ended up with the weight of the lid firmly in its hand vs. having no weight at all. The two exceptions to this are scooping with the small blue cup and opening the bread box. For Cup 1, the models were overly optimistic, with all of the trajectories being classified as succeeding. This could be due to the interactions having more noise than Cup 2 and Parmesan Bottle because of the rigidity of the object. While scooping, Cup 2 and Parmesan do not deform as greatly as Cup 1. For detecting if Curi opened the Bread Box, the models were overly pessimistic with none of the test trajectories being classified as successfully opening the box. We believe that this is because “near-misses” often still pushed on the object and the sensory readings look similar to pushing on the handle successfully. This difference in sensory readings across actions and objects is explored in more detail in Chapter 6.

4.6 Affordance Testing

The next evaluation is a case study of how well existing object-action pairs can classify other object-action pairs within the same affordance. As described in Section 3.5.4, this is an offline approach to testing transfer of object-action pairs within affordances. We present this section to show the limits of using only the previously built object-action pairs to generalize to other interactions from existing object-action pairs. More specifically, whether two different object-action pairs can be correctly identified to be the same affordance. It is interesting to note that for the scoop-able affordance, by using the same action across similar objects, we are in fact simulating how well testing an unseen object could possibly perform.

4.6.1 Experiment Setup

To test if the affordance model can classify an existing object-action pair, we use leave-one-object-out cross validation within an affordance to demonstrate how a robot might test an object for that affordance. This results in three tests for **scoop-able** and four tests for **open-able**. For example, to test if the **Cup1-Scoop** pair would be classified as having the scoop-able affordance, we remove the model learned from the Cup 1 interactions completely and all interactions with Cup 1 then become the test set (10 successful and 10 “near-miss” trajectories). For each test interaction, we use each object-action model within the affordance to evaluate the log-likelihood of that interaction. For scoop-able, this results in 4 different log-likelihood values (from each of the successful and “near-miss” HMMs) and for open-able, 6 log-likelihood values. We label an object as having the affordance if the log-likelihood value is greatest with a successful HMM and not if the value is greatest with a “near-miss” HMM.

Table 4.3: Scoop-able Leave One Object Out

Object-Action Pair	Precision	Recall	F_1	Accuracy
Cup 1-Scoop	0.50	0.10	0.17	0.50
Cup 2-Scoop	0.77	1.00	0.87	0.85
Parmesan Bottle-Scoop	0.59	1.00	0.74	0.65

Table 4.4: Open-able Leave One Object Out

Object-Action Pair	Precision	Recall	F_1	Accuracy
Pastajar-Lift	0.50	1.00	0.67	0.50
Drawer-Pull	0.44	0.89	0.59	0.42
Wooden Box-Push1	0.00	0.00	0.00	0.50
Bread Box-Push2	0.60	0.60	0.60	0.60

4.6.2 Results

The results of affordance testing can be seen in Table 4.3 for scoop-able and Table 4.4 for open-able. However, unlike the results in Section 4.5, it is difficult to fully understand what a “near-miss” example should be classified as given that the interaction was on an object that did indeed *have that affordance*. Instead, it makes more sense to look only at the interactions that successfully found the affordance. This is shown in Fig. 4.6 and Fig. 4.7 with accuracy values for successful interactions reported separately from “near-misses”. We only include the full precision, recall, and F_1 scores in Table 4.3 and Table 4.4 and the accuracy scores of the “near-miss” interactions to show that the models are not merely classifying all interactions as having the affordance.

For scoop-able, the object-action pairs correctly identify an unseen object for both Cup 2 and Parmesan Bottle with accuracy scores of 65% and 85% respectively. Interestingly, Cup1-Scoop does not perform as well. This suggests that the interactions from Cup1-Scoop were not as easily distinguishable, which is supported by our results in Section 4.5. The results of scoop-able show that for an affordance with relatively similar objects, it is possible to identify an unseen object using the learned object-action models. We look next at how well object-action models perform on an affordance that requires very different actions on dissimilar objects. As expected, scoop-able outperforms open-able for identifying an af-

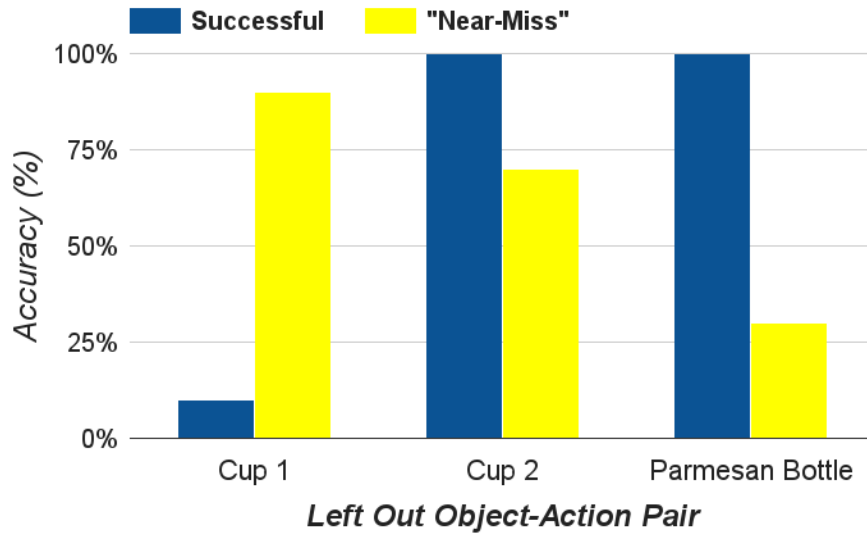


Figure 4.6: Scoopable: Accuracy values for Leave One Object Out. The figure shows the accuracy breakdown between successful and “near-miss” interactions

fordance on unseen objects with an average accuracy score of 67% where open-able has an average accuracy of 53%. As expected, the performance for identifying the open-able affordance on unseen objects is lower than that of scoop-able. Within open-able, the only object that the models could reasonably classify were those of BreadBox-Push2. While the accuracy of PastaJar-Lift and Drawer-Pull are high for successful interactions, it is unclear if it is due to the models truly finding the affordance because all of the interactions (including “near-miss” interactions) were labeled as finding the affordance. For WoodenBox-Push1, the object-action pairs were conservative and did not label any interactions successfully finding the affordance. This could be due to the small F/T values overall felt during the push compared to the other actions that opened objects.

We believe this difference in performance between the two affordances can be attributed to different actions required to find each of these affordances, with open-able requiring more varying actions and scoop-able using similar actions. This suggests that for affordances that require different actions, additional work must be done to adapt and recognize each action (e.g. increasing the number of object-action pairs, including self-exploration, integrating visual information).

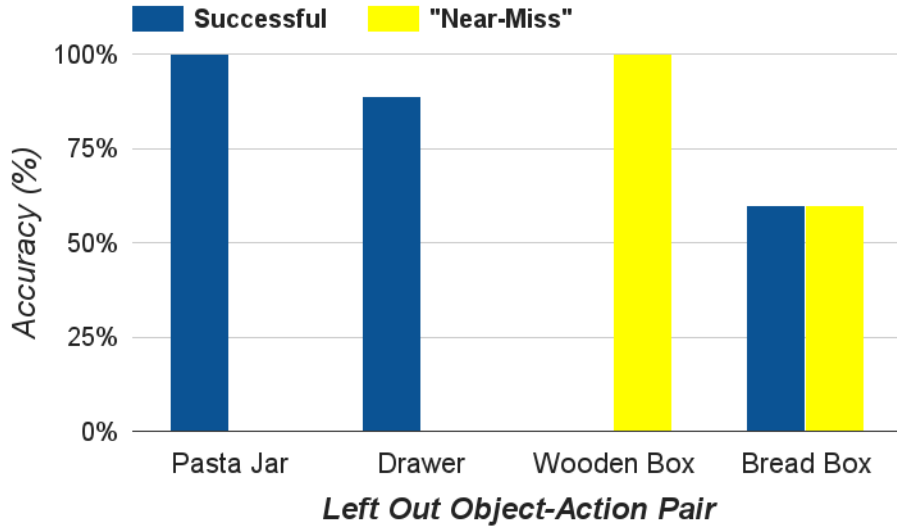


Figure 4.7: Openable: Accuracy values for Leave One Object Out. The figure shows the accuracy breakdown between successful and “near-miss” interactions

4.7 Findings of Learning Haptic Affordances with Human Guidance

The results of both monitoring and affordance testing show that human-guided affordance learning can successfully learn multiple haptic object-action pairs by using LfD and human guidance to initiate the exploration and ground affordances with F/T sensing. For five of the seven object-action pairs, we achieve high F_1 scores at identifying successful execution of the action on the object. We then show that affordance monitoring using multiple object-action pairs can correctly identify the scoop-able affordance on an unseen object with high accuracy. More importantly, our approach allowed us to quickly generate vastly different actions for exploration, which allowed us to analyze and gain insight into affordances not typically explored in the robotics community.

4.7.1 Action Variability

Scoop-able outperforms open-able and we believe this is due to the inherent difference between the affordances open-able and scoop-able. While there are several different methods to open an object, there are far fewer ways to scoop. Furthermore, actions vary signifi-

cantly between the different methods of opening vs. scooping. For example, testing if any object can scoop macaroni, would result in similar sweeping motions with slightly different rotations in the wrist and end-effector offsets (e.g. scooping with a spoon or ladle). However, as seen in this chapter, open-able can break down into several different affordances. One can imagine open-able as comprising of various affordances (e.g. lift-able, pull-able, push-able) while scoop-able is the “lowest” level of the affordance. However, modeling and understanding “high-level” affordances such as open-able is crucial for robots to truly plan and execute tasks. This interesting distinction between the generality and specificity of different affordances is further explored in Chapter 6. Our end goal is for robots to reason about high-level affordances at the task planning stage, but then dive down into the low-level representations of how to achieve that affordance on different types of objects when it comes time to decide how to control the manipulator.

4.7.2 Action Generation

While our current system demonstrates capabilities to test for specific object-action pairs, it cannot adapt to new objects. Later chapters of this thesis tackle this directly by building hybrid control models that use position and haptic feedback to adapt to new objects. This work on action generation fits within the framework of this chapter as it augments the human provided trajectory with robot-generated ones.

4.7.3 Contributions

This chapter showed that in studying haptic affordances, we can begin to understand the role haptics plays in discovering object functions and come closer to building a representation of skills that will allow a robot to achieve tasks in a variety of environments. Furthermore, we have demonstrated that using LfD allows us to quickly provide concrete examples to the robot and allow the robot to discover the “action possibilities” that exist *for the robot* as opposed to any agent. This generated trajectory provides a means to

act to sense the environment and human-guided exploration provides a means to obtain high-quality grounded examples of affordances. The results of this chapter highlight two main pillars of this thesis: the importance of representing affordances with more than just visual data and the benefits of using human demonstrations for affordance learning. The rest of this thesis builds on these core components, and dives deeper into the methods and algorithms that use these core components.

CHAPTER 5

HUMAN-GUIDED ROBOT SELF-EXPLORATION

A major challenge for self-exploration in robot affordance learning is how to determine where to interact with the environment. In Chapter 4, we employed a method of environmental scaffolding and human-guided exploration that required a human teacher to modify the environment. However, requiring a human teacher to modify the environment for the robot is tedious. Furthermore, the human teacher in Chapter 4 was an expert. In this chapter, we look at the questions of *How can we improve this process to require minimum effort from the human teacher?* and *Can naïve human teachers help the robot learn affordances similarly to experts?* To address these questions, this chapter explores different approaches to utilize human guidance and robot self-exploration with naïve human teachers.

We compare three approaches to affordance learning: (1) the traditional **self-exploration** strategy where the robot exhaustively interacts with the workspace; (2) a **human-supervised exploration** strategy where a human provides example object interactions from which the robot learns; and (3) a combined **human-guided** approach that performs self-exploration biased by information provided from human teachers. We compare these three strategies by learning five affordances across four different objects and show that a human-guided approach can learn an affordance model that is as effective as exhaustive self-exploration with an order of magnitude fewer interactions with its environment.

This chapter first explores the result of the exploration algorithms in aggregate across 10 individual teachers. Then the chapter dives into differences between the teachers and analyze the causes for why some users' demonstrations are more informative for exploration than others.

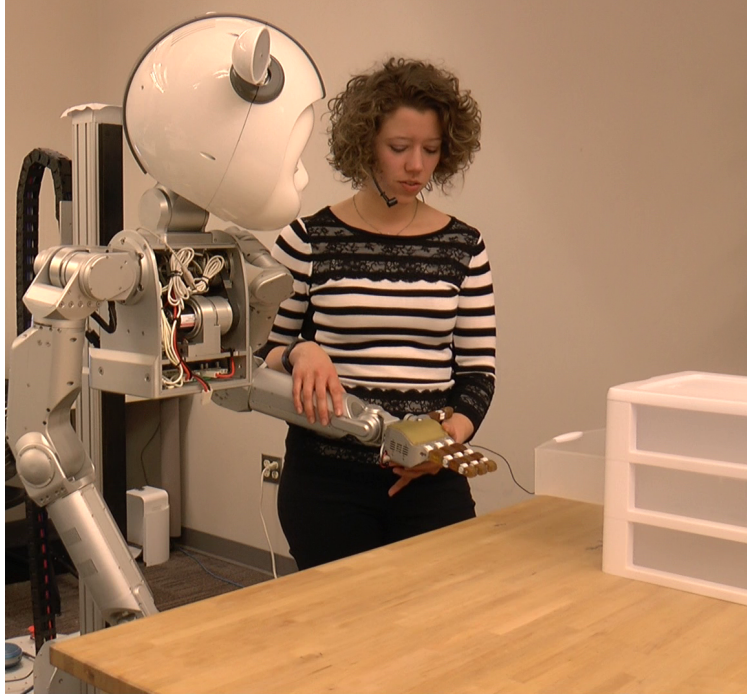


Figure 5.1: A naïve user during the user study teaching “Curi” the robot that the drawer has the open-able affordance.

5.1 Exploration Space for Affordance Learning

As described in Chapter 3, robots need to interact with the environment and observe the effects their actions have on the environment to learn affordances. However, how the robot should interact with the environment is an open question and the space of possible actions the robot can take is infinite. For example, in the simplest case, if the agent’s actions are discrete, it could try all actions on all objects and model the outcomes. However, a better method is required to efficiently sample the infinitely large space of real-world actions the robot could perform to manipulate an object. For example, to open a drawer such as the one seen in Figure 5.1, there are an infinite number of directions a robot could move the drawer in before discovering that it needs to pull the drawer towards itself in a horizontal line. In Chapter 4, we left this exploration process completely up to the expert human teacher by performing environment scaffolding. In this chapter, we consider several novel approaches to reduce the burden on the teacher by adding robot self-exploration.

To make the object exploration tractable using robot self-exploration, we provide the robot with a set of parameterized primitive actions. The exploration space is then defined by the continuous-valued parameters for each primitive action. This choice of representation has gained traction in the reinforcement learning community and has shown great promise with learning actions and skills (Kober et al., 2012; Silva et al., 2014). Consider again the drawer example. Now the opening action can be a primitive defined with three parameters (start, close-hand, and end poses). Note that this still results in a sample space that is infinitely large because these actions parameters are continuous-valued poses of the end-effector. Thus, we present and compare five different strategies in Sections 5.2 and 5.3 for efficiently sampling this space to collect a sufficient set of examples to build object-action affordance models.

For this chapter, we used the same robot “Curi” as in Chapter 4 and seen again in Figure 5.1.

5.1.1 Objects and Actions

We selected four household objects (Figure 5.2) for the robot to interact with. Each of these are tracked using the RGB-D sensor throughout the interaction, from which we record visual object information commonly used in affordance learning (Thomaz and Cakmak, 2009; Montesano et al., 2008) (in 3D space rather than 2D images). We record the color, orientation, volume of the bounding box, the dimensions of the bounding box (x, y, z) , and the squareness of the object (the ratio of the number of points in the object to the area of the bounding box). We also store information from the 6-axis F/T sensor in the wrist (F_x , F_y , F_z , T_x , T_y , T_z) and the robot end-effector (EEF) position relative to the centroid of the object point cloud. This feature vector contains 18 values: 9 (visual), 6 (F/T), and 3 (EEF) and is how we represent the effect of object-action pairs for the affordance learning problem.

The robot can perform two parameterized action primitives: **move** and **pick**. Each is

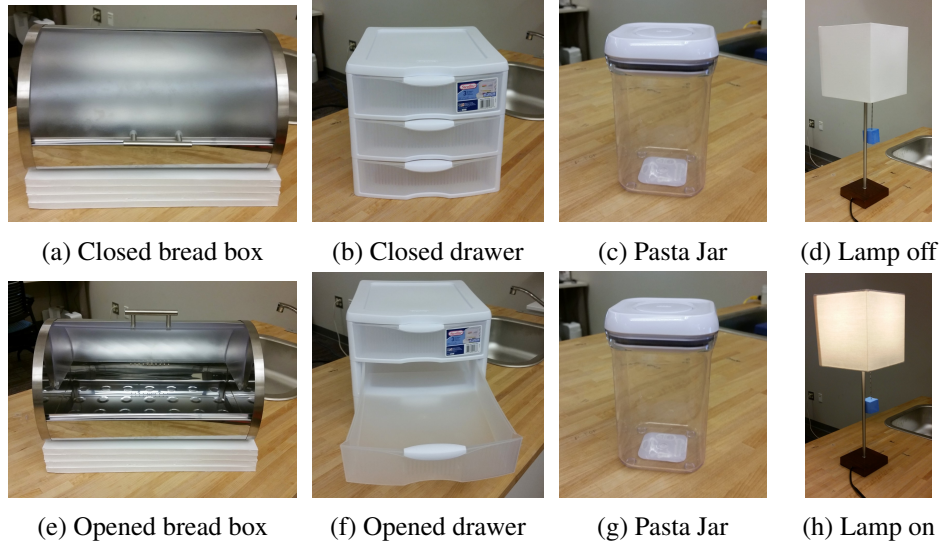


Figure 5.2: Shown are the various objects the robot explored. The top row are the objects before interaction and the bottom row include the same objects with the effect the robot is looking for. Note: pushing the drawer and pasta jar shift the object on the table.

a sequence of EEF poses relative to the centroid of the object point cloud. The EEF pose is the position and orientation of the robot hand for all 6 DOFs. A **move** action has two EEF poses (start and end). The **pick** action has three EEF poses (start, where Curi closes its hand, and end). For both primitives, we generate a trajectory for the EEF by performing a quintic spline between the EEF poses with an average velocity of 1 cm/second. The two actions can be seen in Figure 5.3 where naïve users from our user study are demonstrating **pick** and **move** actions on the lamp and the drawer respectively.

While all poses are needed to define the primitive action, this work will only modify the parameters of the final pose for each primitive action due to the sheer number of object interactions needed to explore the continuous-valued parameters of *all* poses in a primitive action. This is a reasonable simplification since the start pose can be initialized by putting the EEF near the object, as is common in existing affordance work (Fitzpatrick et al., 2003; Hermans et al., 2013b). For both primitive actions in this work, the final pose has the largest impact on successful execution (e.g. the final pose is key in making the move action succeed in pushing an object).

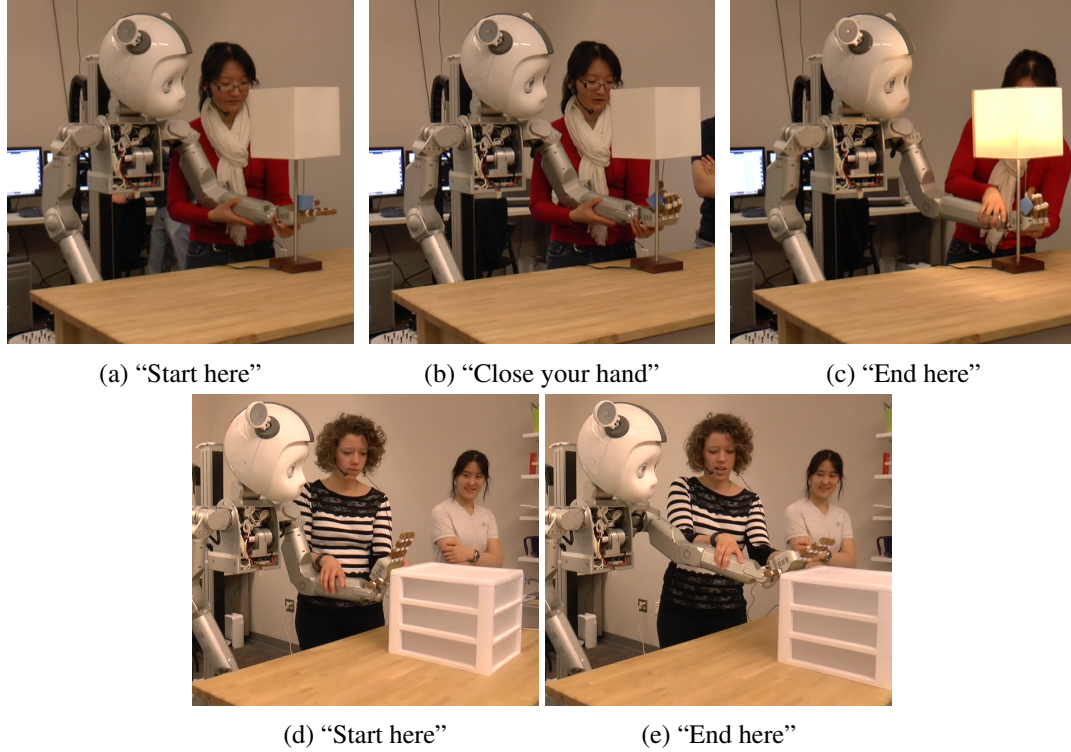


Figure 5.3: Shown are the two primitive actions (**pick** and **move**) taught by users during the user study by using keyframe-based kinesthetic learning from demonstration while the arm is in gravity compensated mode. The users indicate poses within a keyframe using voice commands seen below each image. The **pick** action (5.3a,5.3b,5.3c) consists of a start, close, and end pose and is being demonstrated on the lamp. The **move** action (5.3d,5.3e) consists of a start and end pose and is being demonstrated on the drawer

5.1.2 Affordances

The five specific object-action pairs and their corresponding affordance used in this chapter are described below and summarized in Table 5.1. The effects of each object-action pair can be seen in Figure 5.2. These selected affordances represent a range of difficulty: *simple* affordances that can be found in a large part of the action space during exploration (e.g. push-able can be found in a variety of ways) while *complex* affordances require interacting with the object along a specific dimension of the action primitive space (e.g. open-able on the drawer requires the robot to pull the object towards itself in a particular way, representing a small subset of the object-action exploration space).

- **Bread box:** The lid of the breadbox can be opened with a move action. Affordance:

open-able.

- **Pasta jar:** The pasta jar can be pushed across the table. Affordance: **push-able.**
- **Drawer:** The drawer unit is light enough that the robot can push it across the table. It also contains shelves that can be pulled open. Affordance: **push-able, open-able**
- **Lamp:** When the string attached to the lamp is pulled far enough, the lamp turns on. Affordance: **Turn on-able.**

Table 5.1: Affordances

Object	Action	Effect	Affordance
Breadbox	Move	Moves up	open-able
Pasta jar	Move	Moves	push-able
Drawer	Move	Moves	push-able
Drawer	Pick	Pulls out	open-able
Lamp	Pick	Pulls down	turn-on-able

The question we ask in this chapter is how to best sample the space of our primitive actions’ continuous-valued parameters to interact with objects and collect effective examples for affordance modeling in a way that is *efficient*. We present two baseline approaches, Self-Exploration (SE) and Human-Supervised Exploration (HSE), and compare these to three strategies that represent a *combined* approach: Guided Aggregate Exploration (GAE), Guided Iconic Exploration (GIE), and Guided Boundary Exploration (GBE). All five of these are detailed in the following two sections.

5.2 Baseline Exploration Strategies

Typical self-exploration strategies in robot affordance learning (Fitzpatrick et al., 2003; Stoytchev, 2005; Hermans et al., 2013b) exhaustively sample the space of action parameters. These strategies know only that it should perform actions around the object and the main decisions needed to discretize the space of action parameters relate to (1) what range

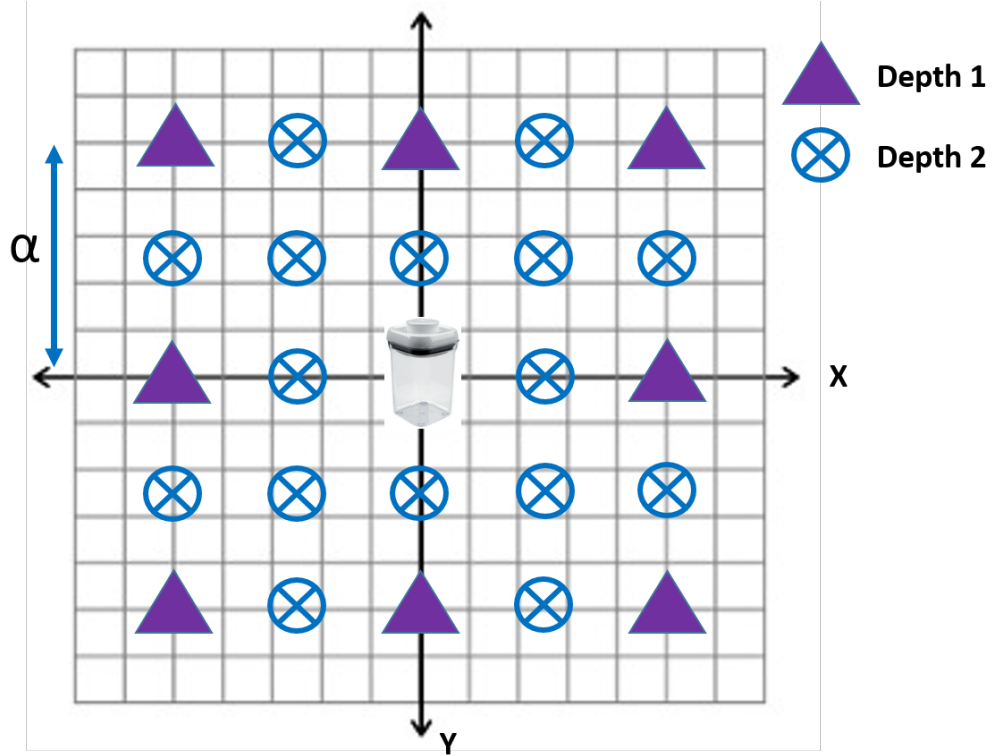


Figure 5.4: Shown above is a visual example of the self-exploration algorithm. The algorithm is viewed in two-dimensions to be visually clear. The exploration is centered around the starting position of the object and the two depths of exploration are shown in two different shapes.

the robot should explore around the object and (2) the resolution (step-size) to use in sampling. We present our version of self-exploration, SE, below (Algorithm 1 and visually in Figure 5.4). To understand the importance of these two variables on exploration, consider the following section.

5.2.1 Self-Exploration (SE)

To represent all 6 DOFs of the EEF, requires three variables (x, y, z) to describe the position and three variables (rx, ry, rz) to describe the orientation in Euler space. However, in practice, it is infeasible for the robot to perform exploration in all six dimensions. For example, suppose we only vary the orientation of the EEF between -90° and 90° , with a step-size of 90° and the position between $-\alpha$ and α with a step-size of α . Assume α

Algorithm 1 Self-Exploration (SE)

```
1: procedure COMPUTEPERMUTATION( $v = [v_0, v_1, \dots, v_n]$ )
2:    $P_{set} \leftarrow$  set of permutations of  $(x, y, z) \forall x, y, z \in v$ 
3:    $P_{set} \leftarrow P_{set} - \{(0, 0, 0)\}$ 
4:   return  $P_{set}$ 
5: procedure GENERATEEXPLORATION(expert dist.  $d$ )
6:    $\alpha \leftarrow d + 10\text{cm}$ 
7:    $D_1 \leftarrow \text{ComputePermutation}([- \alpha, 0, \alpha])$ 
8:    $D_2 \leftarrow \text{ComputePermutation}([- \alpha, -\frac{\alpha}{2}, 0, \frac{\alpha}{2}, \alpha])$ 
9:    $Unique \leftarrow (D_1 \cup D_2) - (D_1 \cap D_2)$ 
10:   $ExploreSet \leftarrow \text{Random}(Unique, 100)$ 
11:  return  $ExploreSet$ 
```

is a constant selected to guarantee the search covers some maximum distance needed for the EEF to have a chance at achieving the object-action pair in question. Even this coarse exploration of the action space results in 676 interactions per EEF pose per object-action pair and, realistically, a higher resolution search will most likely be needed to find the affordance.

It is infeasible for the robot to perform all exploratory actions for all five object-action pairs and all possible primitive action parameter poses. To reduce the number of exploratory actions, we only sample the space of parameters of the final pose of each primitive action. An expert (one of the authors) provides a starting pose (position and orientation) for **move** and a start and close pose for **pick**. These are provided to be *ideal* for achieving the affordance. We believe it is a reasonable assumption to provide the start/close pose, because there exist state-of-the-art algorithms that find the best grasp/interaction points for a wide range of objects (e.g. the handle of the breadbox or the ball on the chain for the lamp). Furthermore, providing this information only helps self-exploration by providing expert information as to where the robot should be interacting. With this assumption, SE exhaustively explores the position (x, y, z) of the final end pose for each primitive action. While varying the orientation could provide additional ways to achieve an affordance, we fix the final pose orientation to be the same as the start pose to keep exploration tractable. Even with these constraints, the number of explorations generated can still result in an

impractical number of exploratory actions. Thus, we limit the number of samples to 100 actions per object.

To generate these 100 samples, we demonstrate one successful interaction with each object to calculate the maximum distance (α) the EEF must travel to achieve each affordance. To ensure this is a conservative estimate we extend the expert demonstrated distance, d , ($\alpha = d + 10cm$), resulting in the maximum distance the SE samples to create exploratory actions. Rather than provide more information to SE about the resolution to sample within these maximum bounds, we adaptively split the action parameter space in half until we reach the designated 100 samples. Thus, we start with a coarse exploration of the space, and continue to sample at a higher resolution until we reach 100 samples of the action space. First, we explore all possible permutations of the three dimensions (x, y, z) for the discrete values: $-\alpha$, 0 , and α . This has 27 different permutations, but we remove the interaction where nothing changes $(0, 0, 0)$ for a total of 26 EEF poses to execute as exploratory actions on the object, which we call D_1 . To sample at a higher resolution, we split the step-size in half, resulting in five discrete values: $-\alpha$, $-\frac{\alpha}{2}$, 0 , $\frac{\alpha}{2}$, and α and a total of 125 permutations. Again, we remove $(0, 0, 0)$ as well as any actions already included in D_1 , resulting in 98 new EEF poses, which we call D_2 . This adaptive split can be seen visually in 2-dimensions in Figure 5.5. To limit each object-action pair to 100 samples, we randomly select 74 interactions from D_2 to add to the 26 interactions of D_1 . Together, D_1 and D_2 compose the exhaustive set of interaction samples for SE.

Note, as mentioned earlier, to make SE tractable, we provided expert information to the algorithm in the form of the start position and orientation of the EEF as well as the maximum distance (α) that the EEF has to explore to find the affordance.

5.2.2 Human-Supervised Exploration (HSE)

The next baseline approach uses a human teacher to fully supervise the collection of examples of object-action interactions. Through action demonstrations, the human teacher

provides successful or unsuccessful examples of the affordance. Our approach, HSE, is the same as described in Chapter 4.

For HSE, we collect data from people in the campus community who had not interacted with our robot before. They used the same two action primitives (**move** and **pick**) that the robot uses during SE. Users teach a **move** action by moving the arm to a start pose and then an end pose, and a **pick** action by moving the arm to a start, grasp, and end pose, which can be seen in Figure 5.3. The robot creates an action trajectory in the same manner as SE, by splining between the action poses. The data used for affordance learning is collected when the robot autonomously executes this human-taught action on the given object. This allows the robot to record the visual and haptic sensory data without erroneously capturing noise from user contact.

We conducted a user study with 10 participants (5 male, 5 female) from a college campus. At the start of their session, participants were instructed briefly on the definition of affordances as well as how to verbally command and move the robot for kinesthetic teaching. For practice, they taught two actions on two objects: lifting the lid off a jar with the **pick** action and tipping an object over with the **move** action. These affordances are not included in our analysis. Once they were comfortable with how to control the robot and had performed several example affordances, we began their real data collection.

The participants taught the robot about the 3 affordances over the 4 objects described in Table 5.1 for a total of 5 object-action pairs. For each object, they were told the specific action (**move** or **pick**) to use and the effect to show the robot. We instructed them to think about what strategy they might use if they were to teach a child about that specific affordance. Participants were instructed during this time to also think about negative examples as a good way to teach a child about an affordance. However, we wanted to see how people teach robots about affordances naturally. Thus, we did not force users to provide negative examples or provide guidance as to how they should teach the robot. A single example for affordance learning was collected each time the robot executes the taught action au-

Table 5.2: Number Examples from Each Exploration Strategy

Object	Action	SE	HSE	GAE	GIE ^a	GBE ^a
Breadbox	Move	100	64	31	12	9
Pasta jar	Move	100	48	30	12	9
Drawer	Move	100	51	31	12	9
Drawer	Pick	96	41	31	12	9
Lamp	Pick	100	51	N/A	N/A	N/A

^a These are the number of examples generated for each user model, since these approaches operate on an individual user basis.

N/A means there were not enough examples for that strategy

tonomously. To generate multiple object-action examples, participants could either move the object and repeat the previous action or they could teach a new action.

For the *complex* object-action pairs (i.e. breadbox-move, drawer-pick, and lamp-pick), participants were given 10 minutes to provide examples to the robot. For the *simple* pairs (pasta jar-move and drawer-move), they were given 5 minutes. The motivation for this difference was based on pilot studies. For simple affordances, users quickly developed strategies for teaching, whereas complex affordances required more time and trials for the user to develop a strategy to get the robot to perform the desired user action. The selected time constraints facilitate the collection of several interactions of each object-action pair and limit each study to within an hour, thus preventing user fatigue. To control for ordering effects in the data, we counter-balanced the order in which the five object-action pairs were taught across users. At the end of the experiment, participants answered a single open-ended survey question that asked them about their teaching strategy. The total number of examples collected across all 10 users can be seen in Table 5.2.

5.2.3 Active Exploration Baselines

While there exist active exploration methods that use human guidance, such as intrinsic motivation (Nguyen and Oudeyer, 2014) and the approach from Nguyen and Kemp (2014), both of these methods require additional information from an expert. Intrinsic motivation

requires a carefully crafted reward signal, which becomes difficult for naïve users to provide. Furthermore, it is unclear how to create reward signals for a wide variety of different motion without carefully hand-tuning each reward signal to each affordance. For example, in this work, pushing, opening the drawer, and turning on the lamp, would all require their own custom reward signals.

The approach from Nguyen and Kemp requires that the initial set of visual features to select from be available *prior* to execution. Specifically, the approach selects the best starting position for the robot to explore and has an expert provide the behaviors (*i.e.* actions) the robots executes after the start has been selected. In this work, we search in the end pose, which is necessary for certain affordances that can vary widely in goal state (*e.g.* pushing an object). As a result, we cannot rely on the visual features provided prior to execution to actively select the next search location. It is important to note that the initialization for the search space (*i.e.* searching around a point with a Gaussian sphere) is similar to the first guided exploration strategy described below (GAE), however, GAE does not assume the shape or size of the Gaussian, and instead learns that from the demonstrations provided by the human teachers.

Due to these limitations for active exploration algorithms, we do not include any active exploration baselines for this work.

5.3 Guided Exploration Strategies

While users provide key information and useful examples of affordances, it is cumbersome to have people provide an exhaustive set of examples for each object-action pair. During self-exploration, the robot can easily generate an exhaustive search, but has no real concept of where to focus that search. Combining the strengths of both approaches should yield the best of both worlds. Our primary research question is how to effectively bias SE with information from human teachers. In this section, we present three novel strategies that differ in how they integrate teacher input for exploration.

5.3.1 Guided Aggregate Exploration (GAE)

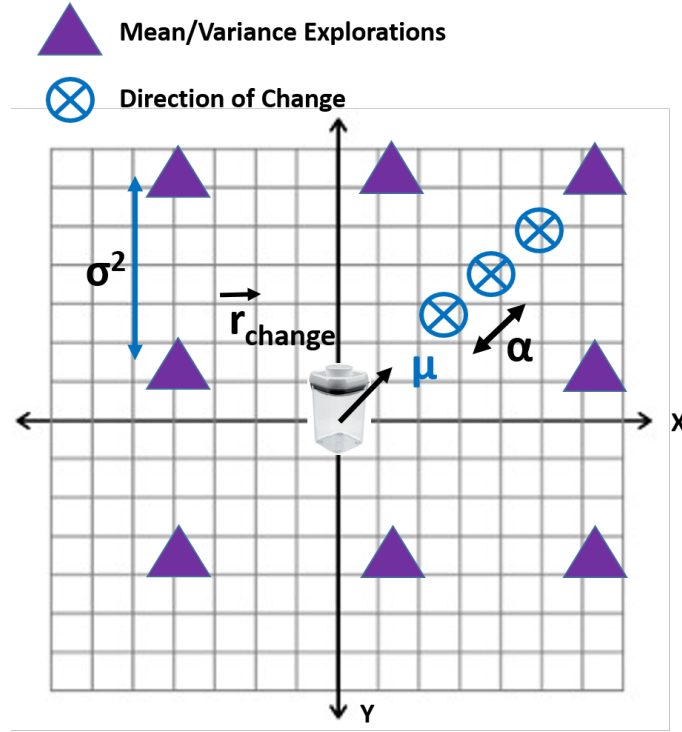


Figure 5.5: Shown is a visual example of the GAE algorithm. The algorithm is viewed in two-dimensions to be visually clear. The exploration is centered around the mean ending position of the first demonstrated by all of the users and exploration is bounded by the variance of the demonstrations.

Our first approach, GAE (Algorithm 2), takes an aggregate view of the guidance that people provided from HSE. The algorithm is described in detail below and shown visually in 2-dimensions in Figure 5.5. For each object-action pair, we build a new set of samples in the action space based on the mean and variance of the *final* EEF position of each first action shown by the ten people in our study. We use only the first action from each user to create a strategy that could be generated using a person’s first intuition for teaching the affordance. However, this is difficult to achieve using just one action primitive and so we built a set that contains the final position of the first action from all users. More concretely, let p^n be the final EEF pose from the first demonstration by user n . Now we define $P_{(j,i)}$ as the set of final EEF positions from all users’ first demonstrations for an object-action pair

Algorithm 2 Guided Aggregate Exploration (GAE)

```
1:  $\alpha \leftarrow$  expert demo dist. +10cm
2:  $p_{(j,i)} \leftarrow \{p^1 \dots p^n\}$  for  $n = 1 \dots 10$ 
3:  $\mu_{ji} = \text{mean}(p_{(j,i)})$ 
4:  $\sigma_{ji}^2 = \text{variance}(p_{(j,i)})$ 
5:  $\vec{r}_{change} \leftarrow \mu_{ji} - EE_{startposition}$ 
6:  $\vec{e}_{change} \leftarrow \frac{\vec{r}_{change}}{\|\vec{r}_{change}\|_2}$ 
7:
8: procedure GENERATEEXPLORATION
9:    $exploreRegions = [\mu_{ji}, \mu_{ji} + \sigma_{ji}^2, \mu_{ji} - \sigma_{ji}^2]$ 
10:   $ExploreSet \leftarrow \text{ComputePermutation}(exploreRegions)$ 
11:   $c \leftarrow 1$ 
12:   $\vec{p}_{change} \leftarrow (0, 0, 0)$ 
13:  while  $\|\vec{p}_{change}\| < \alpha$  do
14:     $\vec{p}_{change} = \vec{e}_{change} * c * \alpha$ 
15:     $ExploreSet \leftarrow ExploreSet \cup \{\vec{p}_{change}\}$ 
16:   $c++$ 
  return  $ExploreSet$ 
```

(o_j, a_i): $P_{(j,i)} = \{p^1 \dots p^n\}$ for $n = 1 \dots 10$. We compute the mean (μ_{ji}) and variance (σ_{ji}^2) of $P_{(j,i)}$, which represents an aggregate of the human provided input, and use them to generate new sample points in the action space. Note that each value contains three numbers (for each axis).

During SE, we sampled the final position of the EEf by adaptively splitting the action space about the starting position using an expert defined α . In GAE, we instead replace α with the computed σ_{ji}^2 and center the sampling of the final position of the EEf using μ_{ji} . This generates an action primitive that starts at the same position defined by the expert and ends using all permutations of the three dimensions (x, y, z) for the discrete values: $\mu_{ji} + \sigma_{ji}^2$, μ_{ji} , and $\mu_{ji} - \sigma_{ji}^2$. For each object-action pair, we have 27 sample locations and use the same EEf orientation used during SE. This strategy explores along the dimensions (x, y, z) of high variance, which are locations in the action space where the object-action can be discovered in a variety of positions. It also constrains the exploration in dimensions of low variance as these are important to finding the affordance.

Additionally, while collecting the SE interactions, we noticed that each object-action pair had a direction of change. For example, the open-able drawer affordance, requires

moving the EEF perpendicular to the drawer towards itself and the open-able breadbox at an angle away from itself. To focus the exploration along this direction of change (\vec{e}_{change}), we do an additional sampling of the EEF action space along this dimension. The \vec{e}_{change} is actually the unit vector between the start (or close) and end positions of the EEF in the action primitive. To explore along this dimension, we scale \vec{e}_{change} by different magnitudes and use the resulting vector as the position in the final EEF pose.

To compute \vec{e}_{change} , we subtract and normalize the expert selected starting position from μ_{ji} . For consistency, we use the same resolution from SE (α) as the base increments to the magnitude. Precisely, $\vec{e}_{change} = \frac{\vec{r}_{change}}{\|\vec{r}_{change}\|_2}$ where $\vec{r}_{change} = \mu_{ji} - EEF_{startposition}$ and the final EEF position is $\vec{e}_{change} * c * \alpha$ where $c = \{1...C\}$. C is the max number of times we can increase the magnitude by before we reach the max exploration distance allowed (set in SE: α). This results in 3 new interactions for pasta jar-move and 4 for all other object-action pairs.

5.3.2 Guided Iconic Exploration (GIE)

Our next approach, GIE (Algorithm 3 and shown visually in Figure 5.6), uses each human teacher’s input individually to bias the exploration of the action space rather than relying on the aggregate of several teachers. Specifically, we use only two samples (the first successful $a^n(S)$ and the first failed $a^n(F)$ interaction) from user n to generate a new set of samples. We select $a^n(S)$ and $a^n(F)$ because this provides crucial information on the location of the boundary between affordance success and failure in the action space. Furthermore, selecting $a^n(S)$ and $a^n(F)$ allows us to determine the viability of having a user provide two samples of the space and having the robot take over afterwards.

We define \vec{r}_{SF} to be the vector extending from S to F , where S is the position (3D) of the EEF in the final pose of $a^n(S)$, and F is the final position of the EEF in $a^n(F)$. The L_2 norm of \vec{r}_{SF} provides a crucial piece of information that, during SE, we had to get from an expert: the exploration resolution the robot should use to achieve the affordance. We can

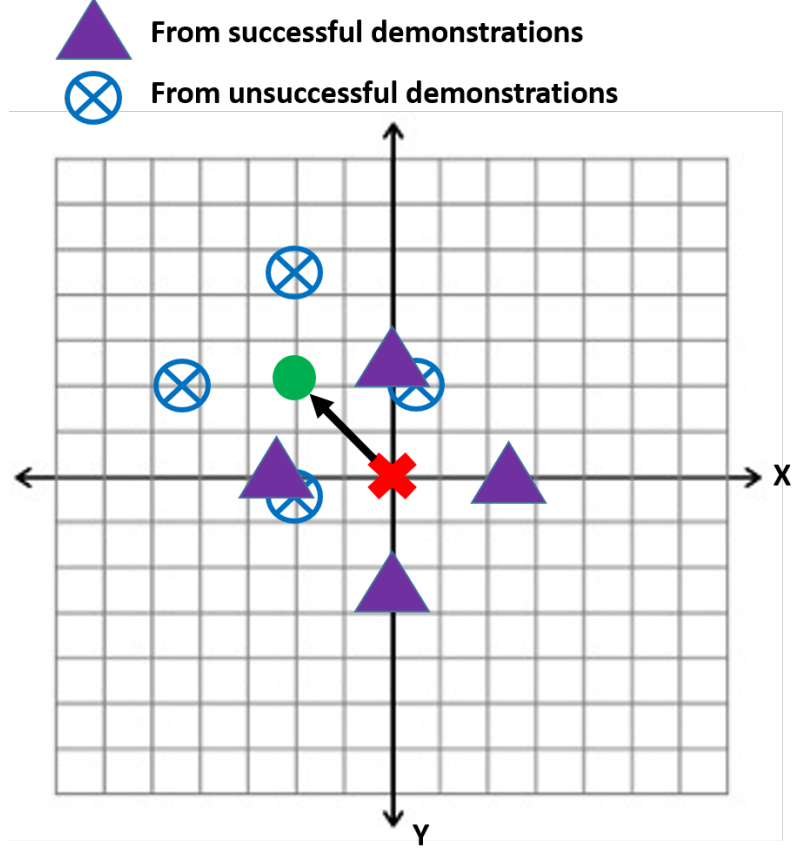


Figure 5.6: Shown is a visual example of the GIE algorithm. The algorithm is viewed in two-dimensions to be visually clear. The exploration uses the first successful and first unsuccessful demonstrations to determine the resolution of exploration as well as where in the space to explore around.

look for the iconic or prototypical examples of successful and failed interactions by adding and subtracting $\|\vec{r}_{SF}\|_2$ from the final pose of the EEF in the action primitive provided by the user in all dimensions (x, y, z) . This results in 6 final EEF poses for $a^n(S)$ and 6 final EEF poses for $a^n(F)$ for a total of 12 final EEF poses. Each of the computed final EEF poses are used to generate primitive actions by replacing the final EEF pose of the primitive action provided by the user.

Note that all poses in the primitive action are generated from the user provided sample. Therefore, not only are we inferring the resolution of the search space with $\|\vec{r}_{SF}\|_2$, but we also no longer need an expert to define the start or close pose of the EEF primitive action. This is particularly important for instances where the a robot manipulator is not standard or

Algorithm 3 Guided Iconic Exploration (GIE)

```
1:  $S \leftarrow$  EEF position of final pose in  $a^n(S)$ 
2:  $F \leftarrow$  EEF position of final pose in  $a^n(F)$ 
3:  $\vec{r}_{SF} \leftarrow (F - S)$ 
4: procedure GENERATEEXPLORATION
5:    $ExploreSet \leftarrow \emptyset$ 
6:   for  $p$  in  $[S, F]$  do
7:      $ExploreSet \leftarrow ExploreSet \cup \{[p_x \pm \|\vec{r}_{SF}\|_2, p_y, p_z]\}$ 
8:      $ExploreSet \leftarrow ExploreSet \cup \{[p_x, p_y \pm \|\vec{r}_{SF}\|_2, p_z]\}$ 
9:      $ExploreSet \leftarrow ExploreSet \cup \{[p_x, p_y, p_z \pm \|\vec{r}_{SF}\|_2]\}$ 
10:  return  $ExploreSet$ 
```

easily modeled, or the object handle is not visually distinct (e.g. the small lip of a drawer).

5.3.3 Guided Boundary Exploration (GBE)

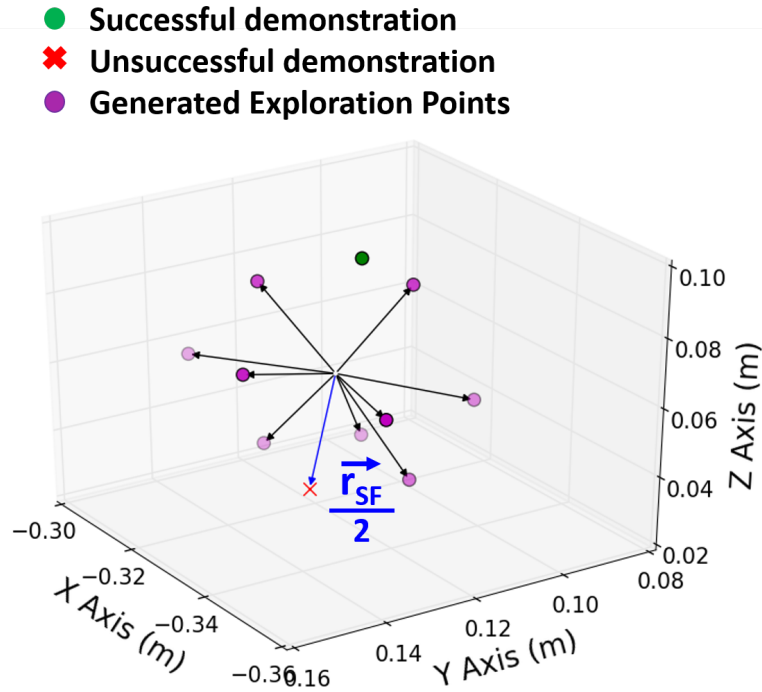
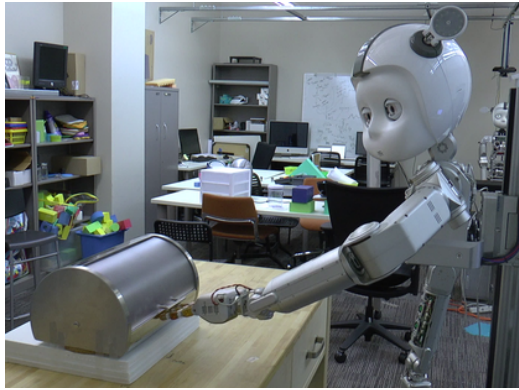


Figure 5.7: Shown is a visual example of the GBE algorithm. The exploration uses the first successful and first unsuccessful demonstrations to determine the resolution of exploration as well as where in the space to explore around.

In GIE, we inferred the boundary between success and fail in the action space by concentrating the new action samples around $a^n(S)$ and $a^n(F)$. Now we introduce GBE (Al-



(a) Approach handle



(b) Moving handle



(c) Open bread box (success)



(d) Approach handle



(e) Open bread box (fail)

Figure 5.8: Shown are example interactions of Curi executing the **move** action on the bread box to find the open-able affordance. The top row (5.8a,5.8b,5.8c) show Curi successfully finding the open-able affordance. The bottom row (5.8d,5.8e) shows an example of Curi failing to find the affordance.

Algorithm 4 Guided Boundary Exploration (GBE)

```
1:  $S \leftarrow$  EEF position of final pose in  $a^n(S)$ 
2:  $F \leftarrow$  EEF position of final pose in  $a^n(F)$ 
3:  $\vec{r}_{SF} \leftarrow (F - S)$ 
4: procedure GENERATEEXPLORATION
5:    $ExploreSet \leftarrow \emptyset$ 
6:   for  $\theta$  in  $[-\frac{\pi}{2}, \frac{\pi}{2}, \pi]$  do
7:      $ExploreSet \leftarrow ExploreSet \cup \{rotateX(S + \frac{\vec{r}_{SF}}{2}, \theta)\}$ 
8:      $ExploreSet \leftarrow ExploreSet \cup \{rotateY(S + \frac{\vec{r}_{SF}}{2}, \theta)\}$ 
9:      $ExploreSet \leftarrow ExploreSet \cup \{rotateZ(S + \frac{\vec{r}_{SF}}{2}, \theta)\}$ 
10:  return  $ExploreSet$ 
```

gorithm 4 and visually in Figure 5.7), which explicitly samples along the boundary. This strategy also uses two action samples ($a^n(S)$ and $a^n(F)$) from each user, and S , F , and \vec{r}_{SF} are the same as before.

To generate the boundary between success and failure in the action space, we use the midpoint between S and F , and coarsely generate multiple vectors circling the midpoint. Specifically, we take $\frac{\vec{r}_{SF}}{2}$ and translate it to the position halfway between S and F . We rotate this new vector about each axis (x, y, z) for the angles $\frac{\pi}{2}$, $-\frac{\pi}{2}$, and π . We hypothesize that one of these vectors is the real boundary for the action space.

GBE generates 9 different final EEF poses in the action space (3 for each axis) that try to find the boundary between the successful and failed affordance interactions. Similar to GIE, we generate each sample by replacing the EEF position in the final EEF pose in $a^n(S)$. Note that since we are using the vector from S to F , we only use $a^n(S)$ and not $a^n(F)$ unlike GIE, which uses both action primitives. Just like GIE, we no longer need an expert for the start pose, close pose, or orientation of the actions primitives. Now these come directly from the human demonstrated action sample.

5.4 Affordance Modeling

We used all five exploration strategies to select actions for the robot to execute to collect example object interactions for all 5 object-action pairs. In total, the robot executed 1219

interactions with the environment (SE (496), HSE (255), GAE (123), GIE and GBE (345)¹).

Each interaction was hand labeled as “Success” or “Failure” depending on whether or not the object interaction achieved the affordance. An example interaction with the breadbox can be seen in Figure 5.8. We used the following cutoffs for “Success”:

- **Breadbox (open-able)** - the breadbox had to be completely open. Any interactions where the robot only opened the box partially is a failure.
- **Pasta jar (push-able)** - the jar is pushed any distance without tipping.
- **Drawer (push-able)** - the drawer is pushed any distance.
- **Drawer (open-able)** - the robot has to pull the drawer out greater than or equal to 5.5 inches (the halfway point)
- **Lamp (turn-on-able)** - the robot has to turn on the lamp without causing the lamp to tip/wobble

To compare the five search strategies, we attempt to train 32 separate models for each object-action pair using the collected data; 2 for strategies that used the holistic approach to search (SE = 1, GAE = 1) and 30 models from the strategies that build a model per user (HSE = 10, GIE = 10, GBE = 10).

Each of these methods take a different approach on how to explore the environment and results in different interactions used to build the affordance model. In essence we look at the efficiency as well as the quality of examples provided by each approach. The overall goal is to determine which strategy provided the “best” interactions for building models of affordances. To determine what is “best”, we evaluate the accuracy of each model at determining if an interaction successfully found an affordance. Specifically, we use binary classification as the method of determining how the types of interactions used to model the

¹GIE and GBE often explored similar locations around the object. As a result, we collected GIE and GBE as a single set and removed similar interactions using a 2cm threshold for position and 45° threshold for orientation.

affordance impact the overall quality of the model at determining success and failure of unseen interactions.

5.4.1 Model Representation

As described in Chapter 3, we represent each object-action object pair using two HMMs. In this chapter, we expand the observation state-space used in Chapter 4 to include visual and EEF information. Specifically, the observation state-space O is composed of visual information, F/T information, and EEF relative to the object as described in Sec. 5.1.1. To select the number of states n for each HMM, we performed 5-fold cross-validation within the training set described in Section 5.4.2. Similar to Chapter 4, for our implementation, we used the Python machine learning library scikit-learn (Pedregosa et al., 2011).

5.4.2 Training and Testing

We split the data collected from each strategy into two sets: train and test. The train set for each strategy contains a randomly selected 80% of the samples from that strategy. The test set is comprised by merging the remaining 20% of the samples from each of the strategies. This results in a test set that contains examples from all strategies. Thus, each strategy trains using 80% of its own sample set, but is tested on a common test set that contains samples from all strategies. We use the standard binary classification metrics of precision, recall, and F_1 as defined in Table 3.1 in Chapter 3.

5.5 Aggregate Results

In this section we focus on the aggregate results across the 10 different teachers. Specifically, we present (1) a characterization of the action space coverage achieved by each exploration strategy (2) the classification performance of the models for each strategy, and (3) qualitative results from the user study survey question.

Table 5.3: Percentage of Positive Interactions Per Strategy

Object-Action	SE	HSE ^a	GAE	GIE ^a	GBE ^a
Breadbox-Move	8%	50%	39%	35%	42%
Pasta Jar-Move	44%	77%	93%	34%	33%
Drawer-Move	43%	78%	100%	28%	36%
Drawer-Pick	18%	38%	74%	33%	44%
Lamp-Pick	2%	4%	N/A	N/A	N/A

^a Values are averaged across each user

Note: Darker shading denotes higher scores and N/A means there were not enough examples for that strategy

5.5.1 Exploration Coverage

We first compare the exploration strategies by the total number and percentage of interactions that successfully achieve the affordances. Seen in Table 5.2, the different strategies result in dramatically different number of samples. HSE resulted in around 5 samples per affordance, whereas SE had 100 samples. By design, all of the Guided strategies fall somewhere between these two extremes. In prior work (Thomaz and Cakmak, 2009), it was shown that self-exploration resulted in mostly negative examples, and their conclusion was that human teachers are good at showing the robot salient positive instances of object affordances. Our data also supports this conclusion. Table 5.3 shows the percentage of successful interactions per affordance. The human teachers (HSE) in our study showed a heavy bias for positive examples, with three of the five affordances having at least half successful examples. This positive bias carries over to the GAE strategy. In terms of coverage of the affordance space, GIE and GBE achieved what we wanted. Biasing SE with supervised examples results in a small number of samples (12 or 9 compared to 100) that have more positive examples than the SE strategy, and more negative examples than HSE.

For the Lamp-Pick affordance, only one of ten users and two SE interactions were able to complete the action successfully due to the arm being too compliant. To train and test a success HMM, we need a minimum of three successful interactions, otherwise the Guided exploration strategies cannot be generated. Thus we exclude Lamp-Pick in the

rest of the results. Furthermore, given the limited data from human teachers, some users did not provide sufficient data to build both HMM models (i.e. min of 3 positive and 3 negative), and in some cases this carried over to the user-biased data sets as well. Column n in Table 5.4 indicates the number of HSE or Guided strategies with sufficient data to build the affordance HMM model.

If we look at the per affordance breakdown in Table 5.3, we can see that aside from the more complex affordances such as Drawer-Pick, users tended to heavily favor providing positive examples. The failures from complex affordances mostly arose from failed attempts in teaching the robot. There does not seem to be a significant difference in positive interactions between Guided Iconic and Guided Boundary.

Coverage can also be evaluated by the physical space the EEF explored and can be visualized by plotting the final position of the EEF relative to the object for successful and failed executions. Fig. 5.9 shows an example visualizing all five strategies for the object-action pair Breadbox-Move. The successful interactions have a high dependency on the x - and z -axis. This makes sense as the EEF must lift the handle away from itself to open the breadbox. In SE, we can see the grid structure of the exhaustive strategy and the large coverage of the action space. The explorations for HSE are highly concentrated and fixated on nearly the same locations in the space. GAE finds examples of successful and failed executions for Breadbox-Move (reflected in Table 5.3), but without a clear action space boundary as in GIE and GBE. GBE examples span a wider range in the action space than GIE.

We can also see that self-exploration looks at a much larger area of the object space for both successful and failed interactions. The human exploration has far more positive examples over a larger area than failed interactions. We can see that Merged exploration is highly concentrated and focused on the boundary between success and failure, which is reflected in the number of positive interactions found per affordance.

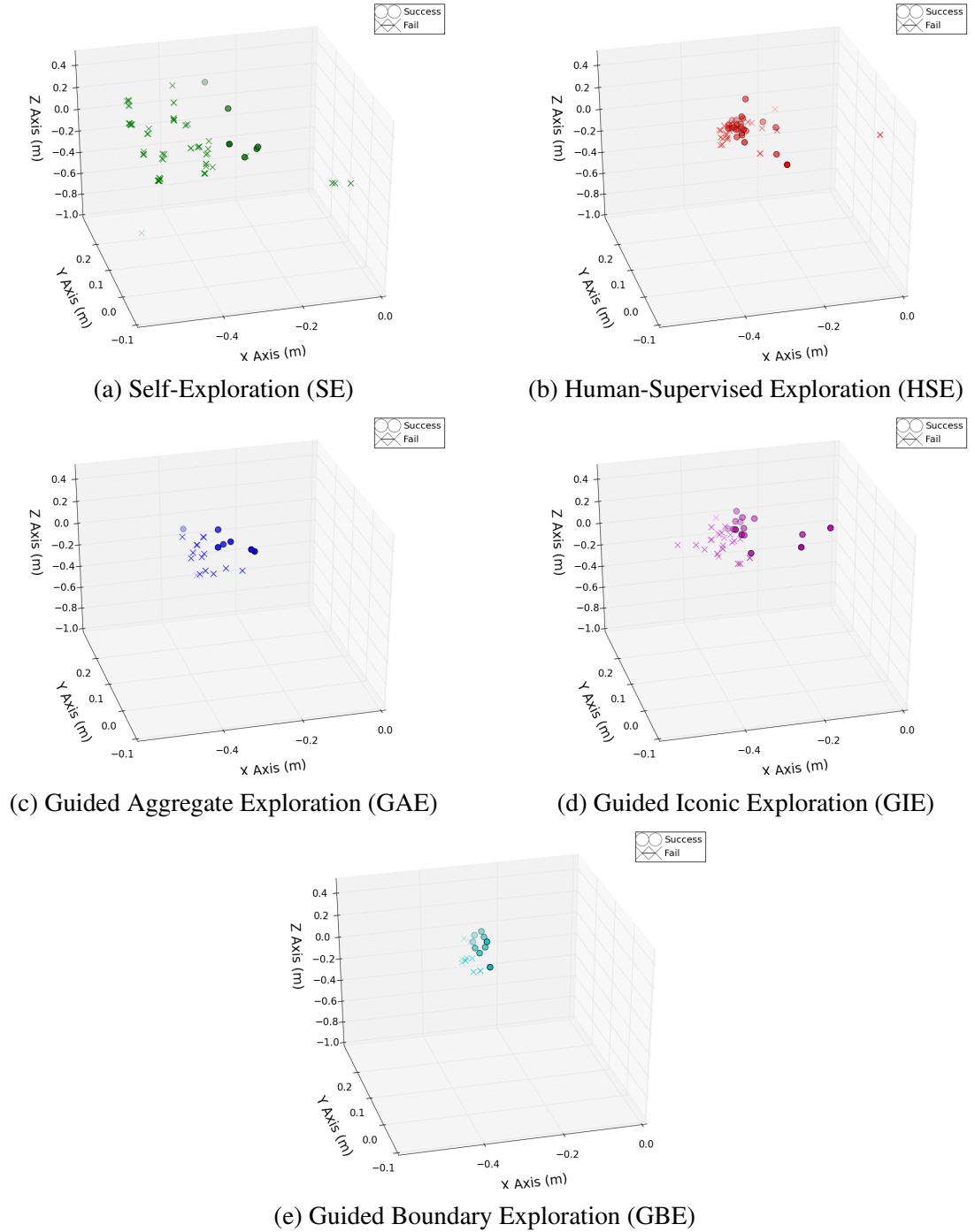


Figure 5.9: The action space (EEF relative to the object) for all five strategies for the affordance Breadbox-Move. Successful interactions are circles and failed interactions crosses. Note: 5.9b, 5.9d, and 5.9e are aggregates over all of the user models.

Table 5.4: Classification Scores on All Exploration Strategies

Aff.	Strategy	n	Precision	Recall	F ₁ Score
Breadbox Move	SE	1	0.73	0.78	0.75
	HSE ^a	10	0.69±0.28	0.48±0.42	0.45±0.34
	GAE	1	0.75	0.53	0.62
	GIE ^a	5	0.75±0.05	0.62±0.35	0.60±0.27
	GBE ^a	3	0.81±0.08	0.53±0.31	0.57±0.24
Pastajar Move	SE	1	0.54	0.97	0.70
	HSE ^a	3	0.90±0.14	0.23±0.24	0.29±0.25
	GAE	0	N/A	N/A	N/A
	GIE ^a	2	0.65±0.12	0.80±0.20	0.69±0.01
	GBE ^a	4	0.53±0.36	0.45±0.40	0.39±0.28
Drawer Move	SE	1	1.00	0.35	0.52
	HSE ^a	3	0.40±0.43	0.06±0.06	0.08±0.07
	GAE	0	N/A	N/A	N/A
	GIE ^a	1	0.51±0.00	0.88±0.00	0.65±0.00
	GBE ^a	1	0.50±0.00	0.56±0.00	0.53±0.00
Drawer Pick	SE	1	0.66	0.93	0.77
	HSE ^a	4	0.66±0.41	0.26±0.42	0.22±0.33
	GAE	1	0.69	0.93	0.79
	GIE ^a	3	0.68±0.01	0.90±0.07	0.77±0.02
	GBE ^a	2	1.00±0.00	0.06±0.03	0.10±0.06

^a Reported values are averaged across the n user or user-biased models.

Note: Darker shading equates to higher scores and N/A means no model could be built using the example

5.5.2 Model Performance

The performance of the models built from each strategy is shown in Table 5.4. For exploration strategies that resulted in individual user or user-biased models (Supervised, Guided Iconic, Guided Boundary), we report the mean and variance of all the models within that strategy. HSE has the worst performance. This could be due to the fact that overall, users tended to overly focus on positive interactions and that these models were built from the least amount of data (5 examples compared to 9, 12, 30 or 100). These results support our hypothesis that the limited data we can collect from a person in 5-10 minutes of interaction is not sufficient to build models on par with exhaustive self-exploration consisting of 100 interactions with an object.

Next we turn to the question of whether or not the Guided strategies can help bridge this performance gap between SE and HSE. Our results show that both GIE and GAE achieve higher performance than GBE and approach the performance of SE with significantly less data. GIE outperforms GAE in two ways: (1) across the four affordances, GAE only generates 2 working affordance models (due to the focus on successful examples) whereas GIE generated 11 (seen in column n in Table 5.4) and (2) GIE is likely to be the more practical approach compared to GAE since it can be used with a single individual as opposed to requiring data from multiple teachers. On average, the Guided Iconic exploration strategy reaches similar performance levels as the exhaustive Self-Exploration strategy. This is true for every object-action pairs except for Breadbox-Move. For the pair Breadbox-Move, we can look closer at the precision and recall scores and see that Guided Iconic has similar precision scores, but is more conservative at labeling the breadbox with the open-able affordance.

Surprisingly, the simple affordances (Pastajar-Move & Drawer-Move) performed worse on average across the strategies than the complex affordances. The only exception is GIE for Pastajar-Move, which slightly outperforms Breadbox-Move. One possibility for the discrepancy between these affordances could be related to how affordances are not really on a binary spectrum, but rather there are varying levels (e.g. slightly push-able vs. very push-able). This suggests that the task should be a regression task where we label the affordances with values (e.g. 1cm vs. 5cm). In our case, we set hard cutoffs for judging the success of each affordance. Successful interactions were more obvious for complex tasks (e.g. drawer or breadbox fully opening) compared to simple tasks (e.g. shifting the pasta jar or drawer 10 cm vs. 1 cm across the table; both of which were considered successful interactions).

Finally, it is important to note the high variance for exploration strategies that generated models per user. While the average GIE performance is similar to SE, many individual models achieved performance that surpassed the SE models, with an order of magnitude

fewer examples. This shows that certain users provided better examples than others, and future work is to understand how to accomplish this with all users.

5.6 User Specific Results

To understand why some user specific models performed better than others, we take a deeper look at each user. We first look at the precision, recall, and F_1 scores for each user specific strategy (HSE, GIE, and GBE). This can be seen in Table 5.5. As described previously, several users outperformed self-exploration (5 users in HSE, 5 users in GIE, and 1 user in GBE across all four object-action pairs). Table 5.5 also shows why GIE outperforms GBE and HSE on aggregate. While many of the models from HSE and GBE do equally as well as models from GIE, there are several models in HSE and GBE that perform poorly. In contrast only 1 model in GIE performs poorly (Breadbox-Move: User 8).

Interestingly, for some users, even though the user provided enough positive or negative examples to build models for HSE, GIE and GBE were not able to find enough examples. This tells us that the first successful and unsuccessful demonstrations were not diverse enough to provide a sufficient amount of exploration range for GIE or GBE. Specifically, for all object-action pairs except for Pastajar-Move, there were users where HSE could build a model, but GIE or GBE could not. To better understand this, we looked at the pose of the first successful and unsuccessful demonstration provided from users that generated good GIE/GBE models and compared them to users who did not. In Figure 5.10, we can see the comparison of the final pose of each user’s first successful and unsuccessful demonstration. Visually, the general location of the demonstrations seem relatively similar. However, computing the average Euclidean distance between the successful and unsuccessful positions (shown in Table 5.6), shows that participants who provided demonstrations that were further apart in distance, allowed GIE and GBE to generate better models. This makes intuitive sense as both GIE and GBE rely on the user’s demonstration to determine the res-

Table 5.5: User Specific Classification Scores for Strategies HSE, GIE, and GBE

Object-Action	User	Human			Iconic			Boundary		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Breadbox Move	User 1	1.00	0.01	0.03	N/A	N/A	N/A	N/A	N/A	N/A
	User 2	0.74	0.29	0.42	0.69	0.62	0.65	0.71	0.90	0.79
	User 3	0.71	0.84	0.77	N/A	N/A	N/A	N/A	N/A	N/A
	User 4	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A	N/A
	User 5	0.60	0.13	0.22	N/A	N/A	N/A	N/A	N/A	N/A
	User 6	0.72	0.97	0.82	0.72	1.00	0.84	0.83	0.57	0.68
	User 7	0.73	0.99	0.84	0.72	0.99	0.83	N/A	N/A	N/A
	User 8	1.00	0.09	0.16	0.80	0.06	0.11	0.90	0.13	0.23
	User 9	0.73	0.97	0.83	0.82	0.46	0.58	N/A	N/A	N/A
Pastajar Move	User 1	1.00	0.09	0.17	0.77	0.61	0.68	0.65	0.67	0.66
	User 2	0.70	0.58	0.63	N/A	N/A	N/A	0.00	0.00	0.00
	User 3	1.00	0.03	0.06	0.53	1.00	0.69	1.00	0.15	0.26
	User 6	N/A	N/A	N/A	N/A	N/A	N/A	0.48	1.00	0.65
Drawer Move	User 2	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A	N/A
	User 3	1.00	0.03	0.06	N/A	N/A	N/A	N/A	N/A	N/A
	User 5	N/A	N/A	N/A	0.51	0.88	0.65	0.50	0.56	0.53
	User 8	0.21	0.15	0.17	N/A	N/A	N/A	N/A	N/A	N/A
Drawer Pick	User 3	0.65	0.98	0.78	N/A	N/A	N/A	N/A	N/A	N/A
	User 4	N/A	N/A	N/A	0.69	0.84	0.76	1.00	0.02	0.04
	User 5	1.00	0.02	0.04	0.67	0.87	0.76	N/A	N/A	N/A
	User 6	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A	N/A
	User 7	N/A	N/A	N/A	0.67	1.00	0.80	1.00	0.09	0.16
	User 8	1.00	0.02	0.04	N/A	N/A	N/A	N/A	N/A	N/A

Note: Darker shading equates to higher scores and N/A means no model could be built using the example

olution to search within. If the resolution is too small, then the algorithm does not explore a large enough range to capture a balanced set of positive and negative interactions.

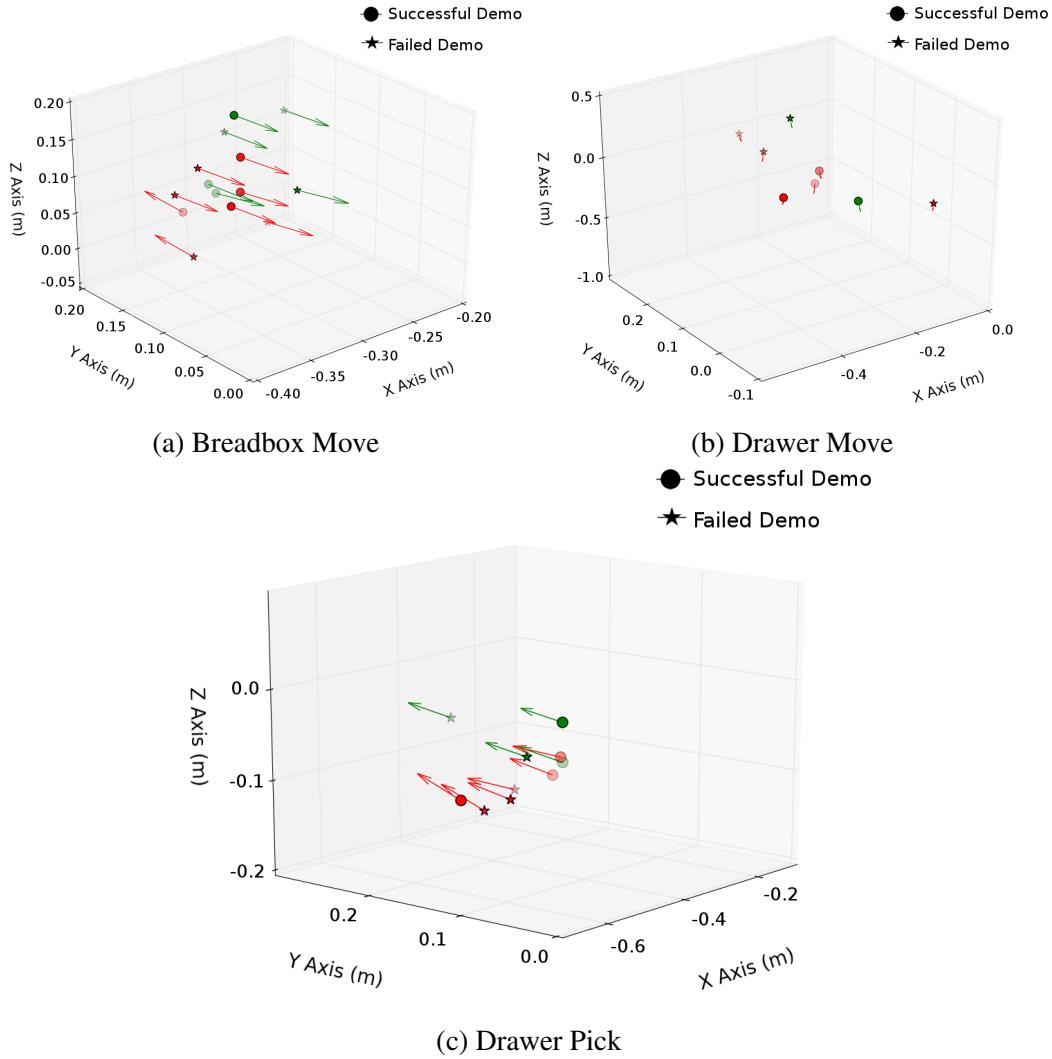


Figure 5.10: This shows the first successful and unsuccessful demonstration for the affordances Breadbox-Move, Drawer-Move, and Drawer-Pick. The symbols indicate if the demonstration given was a success or fail demonstration. The colors separated the users that could generate GIE/GBE models and those who could not (green - models were generated, red - models were not). Note: this figure requires color to fully understand.

5.6.1 Exploration Coverage

In 5.5.1, we visually showed the difference between strategies in the action space (EEF position relative to the object). Here we visualize the action space to provide insight into

Table 5.6: Average Distance Between Demonstrations

Object-Action	Good (cm)	Poor (cm)
Breadbox Move	8.46 ± 0.05	7.3 ± 0.19
Drawer Move	32.94 ± 0.0	24.60 ± 0.14
Drawer Pick	8.98 ± 0.01	8.08 ± 0.019

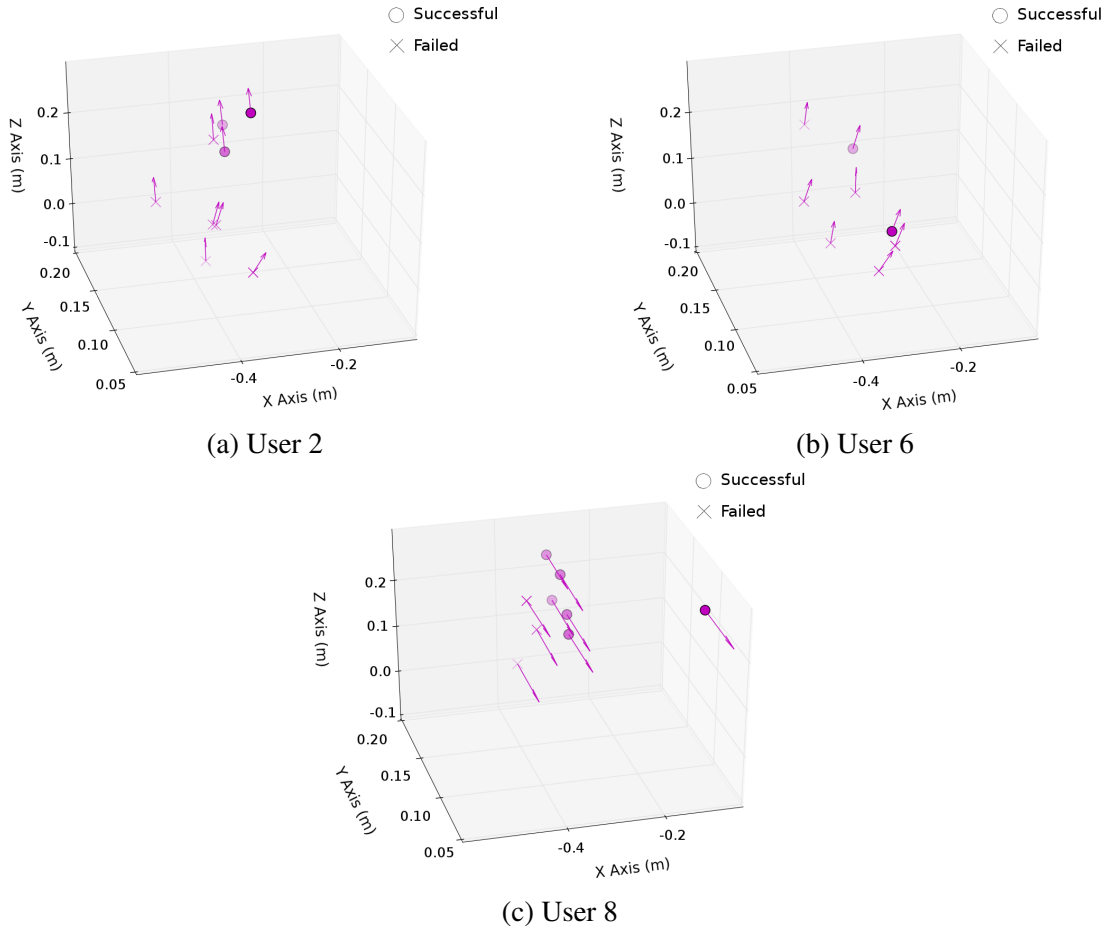


Figure 5.11: Shown is the action space (EEF relative to the object) for strategy Guided Iconic Exploration for users 2, 6, and 8 for the object-action pair Breadbox-Move. The arrow indicates the direction the EEF palm is facing. Successful interactions are circles and failed interactions are crosses.

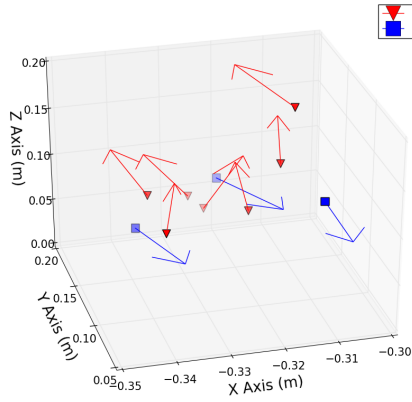
why a model based on a certain user’s demonstrations might not have generated a good model. Table 5.5 shows that only one user’s GIE model performed poorly. Furthermore, this particular user (user 8) does poorly across all of the user specific strategies (HSE, GIE, GBE) for the object-action pair Breadbox-Move. The next set of graphs will be presented as a case study to determine what differences exist between user 8 and the rest of the users.

In Figure 5.11, we see three different users’ exploration points generated for the object-pair Breadbox-Move for the strategy GIE. Circles represent successful interactions while crosses represent failed interactions. The orientation of the palm of the EEF is also shown as a vector. The figure shows that the orientation of the EEF played a clear role in differentiating user 8 from the rest of the users. User 8 chose a different orientation when opening the bread box and video verifies that user 8 had Curi’s palm facing down as opposed to up to lift the handle.

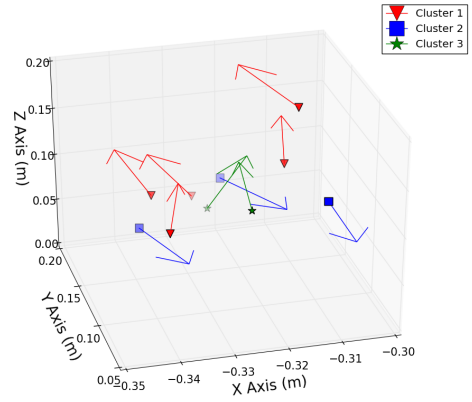
This suggests that the demonstration from user 8 was not bad, but rather *different* from the other demonstrations provided by other users. Furthermore, we hypothesize that if there existed a subset of the evaluation set that is similar to the demonstrations from user 8, then the performance for that user would increase. To understand and determine which users were most similar to each other, we take a simple approach of clustering all of the user’s first demonstrations (those used to seed GIE and GBE) using a standard unsupervised clustering algorithm k-means.

5.6.2 Clustering

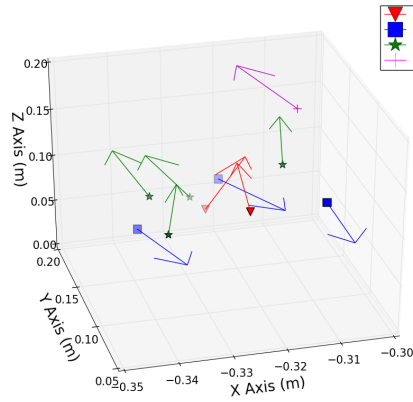
To cluster the demonstrations using k-means, there are two decisions that need to be made (1) what metric to use for distance between demonstrations and (2) the number of clusters we expect to see. We chose to use the Euclidean distance of the EEF position and orientation of the palm relative to the object. This decision allows us to focus on what the user demonstrated relative to the object without looking at the effects generated by the demonstration. Furthermore, Euclidean distance is a natural metric between points in three-



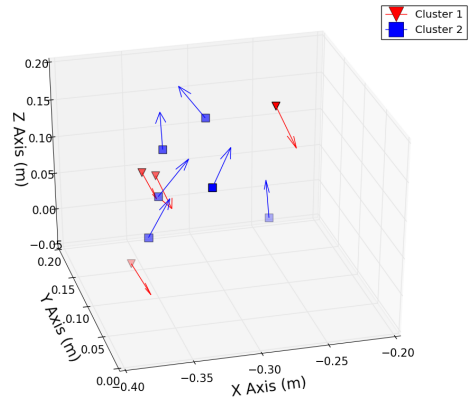
(a) Success; Size: 2



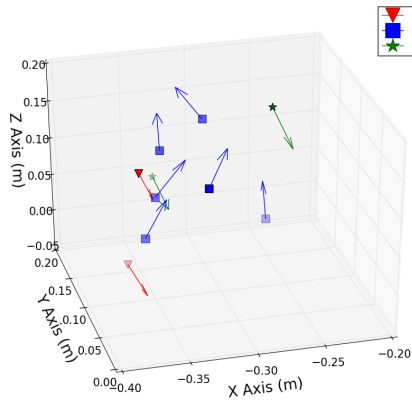
(b) Success; Size: 3



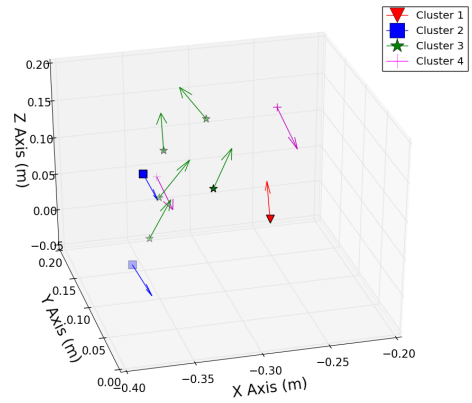
(c) Success; Size: 4



(d) Fail; Size: 2



(e) Fail; Size: 3



(f) Fail; Size: 4

Figure 5.12: Displayed are the first success and fail demonstration from each user in action space (EEF relative to object) for different clusters sizes (2,3,4). The arrow indicates the direction the EEF palm is facing. Note: this figure will be easier to decipher with color.

dimensions. While orientation of the EEF is stored as a quaternion, when computing the distance between demonstrations, orientation is represented by the normalized unit vector of the direction the palm of the EEF is facing (shown in Figure 5.11).

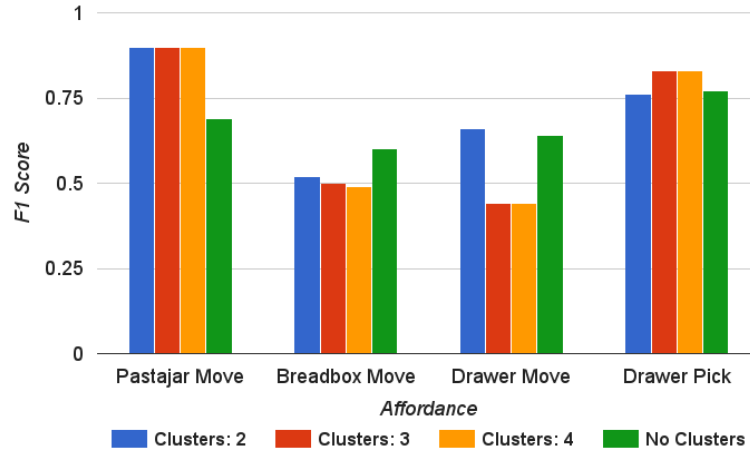
We chose several cluster sizes and compared user performance within clusters. As a reminder, we use both the first successful and first unsuccessful demonstration from each user when generating exploration points. We cluster successes and failures separately. While successful demonstrations are (typically) intentional, failures are not guaranteed to be intentional. Often during the HSE, the human-user’s first failure was a result of failing to demonstrate a successful interaction.

We visually show which cluster the user’s first demonstrations (successes and failures) fall into for the object-action pair Breadbox-Move, which can be seen in Figure 5.12. For clusters of size two, it seems that orientation of the EEF plays a larger role in cluster membership. As the cluster sizes increase, position plays a larger role.

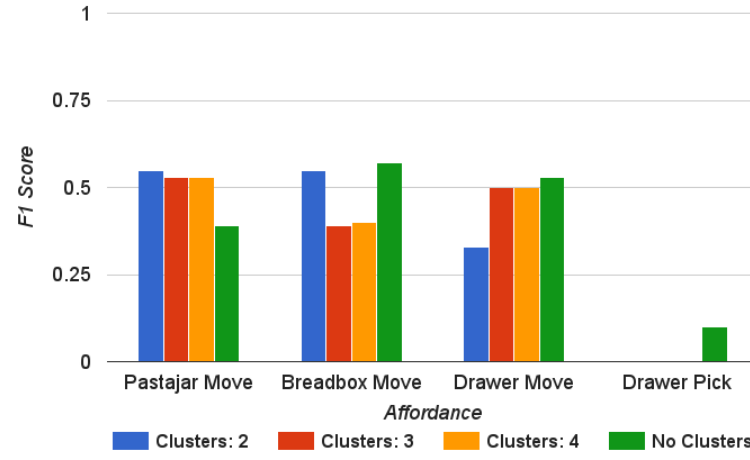
5.6.3 Clustering Performance

To verify that the difference in initial demonstrations impacts the final performance of a user-specific model, we hypothesize that there exists a subset of robot interactions that are similar to the user and the model would perform well on this subset. We generate the user-specific test set by taking a portion of the original test set (20% of each strategy). This subset is determined based on the cluster membership of the user-specific model. For example, to generate the test set for user 1, we first determine what cluster generated from k-means the user falls into (for both success and fail). Then the test interactions associated with all users in that cluster are pulled and these interactions make up the test set for user 1. Note, this means that the original test set does not contain any interactions that are not associated with a specific user (i.e. only test interactions from GIE, GBE, and HSE are used).

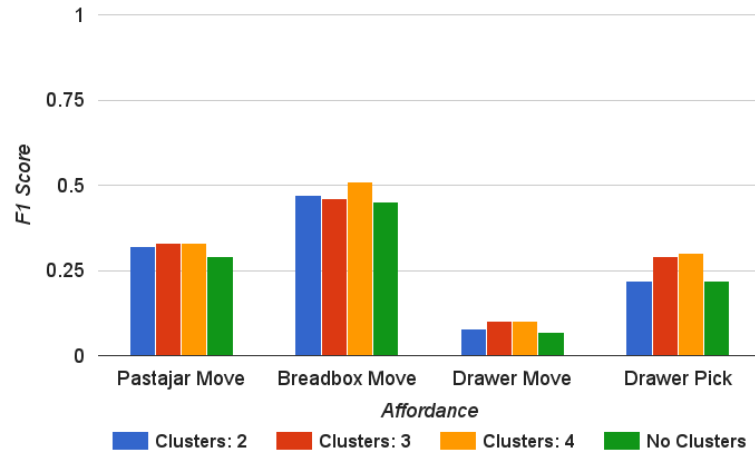
The aggregate performance of each user-specific model for each object-action pair for



(a) GIE



(b) GBE



(c) HSE

Figure 5.13: Aggregate F_1 values across users and affordances and clusters sizes.

all cluster sizes are shown in Figure 5.13. The average F_1 scores are shown in comparison to the original aggregate user-specific scores for GIE and GBE. Overall, with the exception of Pastajar-Move, selecting a subset of the test set based off of the user’s first demonstration was unable to improve the performance of the user-specific models. As expected, there does not exist a single cluster size that is favored across object-action pairs. We believe this is due to the inherent differences in each pair (i.e. each pair has its own subset of unique interactions).

Given that clustering did not work uniformly across all object-action pairs, we only present detailed user-specific scores for Pastajar-Move and Breadbox-Move to understand why some aggregates went up while others went down. This can be seen in Table 5.7. Overall the models generated from HSE did not drastically change. This is likely because HSE models have access to all demonstrations provided within a single user whereas GIE and GBE are limited to the first successful and first failed demonstration. This is amplified by users going out of their way to provide different and interesting demonstrations for each pair during the user study. Looking at Pastajar-Move, on the aggregate level, clustering the test set improves the overall performance. On an individual level, the models that were already performing well improved and the models that were not forming well either dropped or didn’t change. This is seen for several users in Breadbox-Move as well. While it is not surprising that existing high-performing models improve when looking a subset of interactions that are most similar to it, it is surprising that the models that were performing poorly, performed even worse. This suggests that there is something else occurring within the effect space of the affordance that we are not capturing by clustering the EEf pose relative to the object. We explore this in Section 5.6.5.

5.6.4 Ratio of Success and Failure

Before we look into the observation space of the learned HMM, we look at one final metric presented in previous results (Table 5.3), the ratio between successful and unsuccessful

Table 5.7: User Specific Classification Scores for HSE, GIE, and GBE After Clustering for Breadbox-Move and Pastajar-Move

		Human			Iconic			Boundary		
Object-Action	User	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Breadbox Move	User 1	1.00	0.02	0.04	N/A	N/A	N/A	N/A	N/A	N/A
	User 2	0.62	0.33	0.43	0.42	0.54	0.47	0.48	0.83	0.61
	User 3	0.56	1.00	0.72	N/A	N/A	N/A	N/A	N/A	N/A
	User 4	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A	N/A
	User 5	1.00	0.18	0.31	N/A	N/A	N/A	N/A	N/A	N/A
	User 6	0.66	0.95	0.78	0.67	1.00	0.80	0.80	0.64	0.71
	User 7	0.86	1.00	0.92	0.88	0.96	0.92	N/A	N/A	N/A
	User 8	1.00	0.14	0.24	0.50	0.02	0.04	0.67	0.05	0.09
	User 9	0.68	1.00	0.81	0.76	0.43	0.55	N/A	N/A	N/A
Pastajar Move	User 1	1.00	0.15	0.27	0.83	0.77	0.80	0.85	0.85	0.85
	User 2	0.80	0.62	0.70	N/A	N/A	N/A	0.00	0.00	0.00
	User 3	0.00	0.00	0.00	0.37	1.00	0.54	1.00	0.15	0.27
	User 6	N/A	N/A	N/A	N/A	N/A	N/A	0.74	1.00	0.85

Note: Darker shading equates to higher scores and N/A means no model could be built using the example

Table 5.8: Percentage of Positive Interactions Per User For Strategies HSE, GIE, and GBE

Object-Action	User	HSE	GIE	GBE
Breadbox Move	User 1	0.50	0.11	0.17
	User 2	0.40	0.33	0.40
	User 3	0.33	0.11	0.17
	User 4	0.60	0.50	0.50
	User 5	0.50	0.10	0.14
	User 6	0.38	0.25	0.43
	User 7	0.50	0.29	0.67
	User 8	0.50	0.67	0.67
	User 9	0.50	0.44	0.20
	User 10	0.75	0.12	0.12
Pastajar Move	User 1	0.50	0.50	0.43
	User 2	0.60	0.20	0.67
	User 3	0.60	0.50	0.67
	User 4	0.75	N/A	N/A
	User 6	0.67	0.12	0.33
	User 7	0.67	0.12	0.20
	User 8	0.67	N/A	N/A
	User 10	0.75	N/A	N/A
Drawer Move	User 1	0.75	0.14	0.25
	User 2	0.60	0.11	0.14
	User 3	0.67	0.89	0.88
	User 4	0.67	N/A	N/A
	User 5	0.75	0.25	0.33
	User 8	0.50	0.11	0.14
	User 10	0.67	N/A	N/A
Drawer Pick	User 3	0.50	0.10	0.14
	User 4	0.33	0.33	0.67
	User 5	0.50	0.33	0.67
	User 6	0.50	0.12	0.20
	User 7	0.67	0.22	0.67
	User 8	0.50	0.14	0.20
	User 9	0.67	N/A	N/A
	User 10	0.25	0.75	0.67

Note: Darker shading equates to higher scores and N/A means no model could be built using the examples (recall we need a minimum of 3 examples each of success and failure to build models).

interactions. Previously, we concluded that overall GBE and GIE had a more balanced set of positive vs. negative examples compared to HSE or self-exploration. We now take a look at this ratio on a per user basis. Results are summarized in Table 5.8.

Similar to Table 5.5, we only show the users that had enough positive and negative interactions to build a HMM. As mentioned earlier, we need a minimum of 3 examples of both positive and negative to build a model. For the models that could be built, we can look back at the detailed results in Table 5.5, and pull out specific users that performed well and performed poorly. For strategy GIE for the object-action pair Breadbox-Move, users 2, 6, 7, and 9 outperformed user 8. We can see that user 8 has a much higher percentage of positive executions than these other users. This is also true for the strategy GBE, where user 8 performs poorly compared to 2 and 6. This trend is consistent across the rest of the pairs as well: users with a particularly high number of success demonstrations have models that perform poorly.

Digging deeper, we discovered that for some of the models that performed poorly, they essentially classify everything as not having the affordance, resulting in low recall and non-existent precision values. This indicates that when users have too many examples of successful interactions we cannot build a good HMM representing the expected effects of these successful interactions. We believe this happens due to the nature of the F/T data we are using to represent the effects of affordances. Recall that our exploration strategies are taking a couple of human demonstrations as seed examples and then varying these slightly in the end-effector space to get several new examples *around* the original ones. However, even though a slight change in position of the end-effector to the object may still result in a successful interaction (e.g. the breadbox still moves), it can drastically change the signal seen on the F/T plate at the robot’s wrist. Thus, a dataset that includes a large number of successful examples is more likely to be highly varied. It is difficult to build a model that represents all of these different effects at once. On the other hand datasets with a limited number of success interactions, are more likely to only include a single *way* of achieving

the affordance, that is more consistent in the sensory space and easier to model.

5.6.5 Affordance Effect Space

In Section 5.6.3 and Section 5.6.4 we have seen evidence that the effect space of the object-action pairs play large role in the quality of models built. To understand why, we take a deeper look at the multivariate Gaussian distribution that represent the observation space of the learned HMMs. We choose to look at the last state because achieving an affordance is highly dependent on the final pose of the interaction. This also allows us to focus on a specific snapshot in time where the effect of the affordance is most likely to have occurred as opposed to the entire trajectory of the interaction.

Recall we have an 18-value feature vector that represents the observation state-space of the HMM. To focus on the specific dimensions that have the greatest change in the effect space, we perform principal component analysis (PCA) on the observation space. Concretely, we compute the principal components of the set of all mean values of the multivariate Gaussian distribution for all of the generated HMMs. This is done specific to the set of successful HMMs and unsuccessful HMMs. This results in two transforms - one for each set of means. We selected the top three principal components, which account for 99.9% of variance for both sets of HMMs (successful and unsuccessful). We compare this reduced set of components from user models that produced good object-action pairs vs. those who did not. As a reminder, successful HMMs were HMMs generated from robot trials that successfully found the affordance whereas unsuccessful HMMs were generated from trials that did not. Furthermore, good user models (or good HMMs) are models that performed well at classifying unseen interactions whereas poor user models (or poor HMMs) did not score well in classifying new unseen interactions.

Figure 5.14 shows the different principal component values for each individual model for the object-action pair, Drawer-Move. The green bars show the users that had good HMMs and the red bars show users with HMMs that performed poorly. While the means

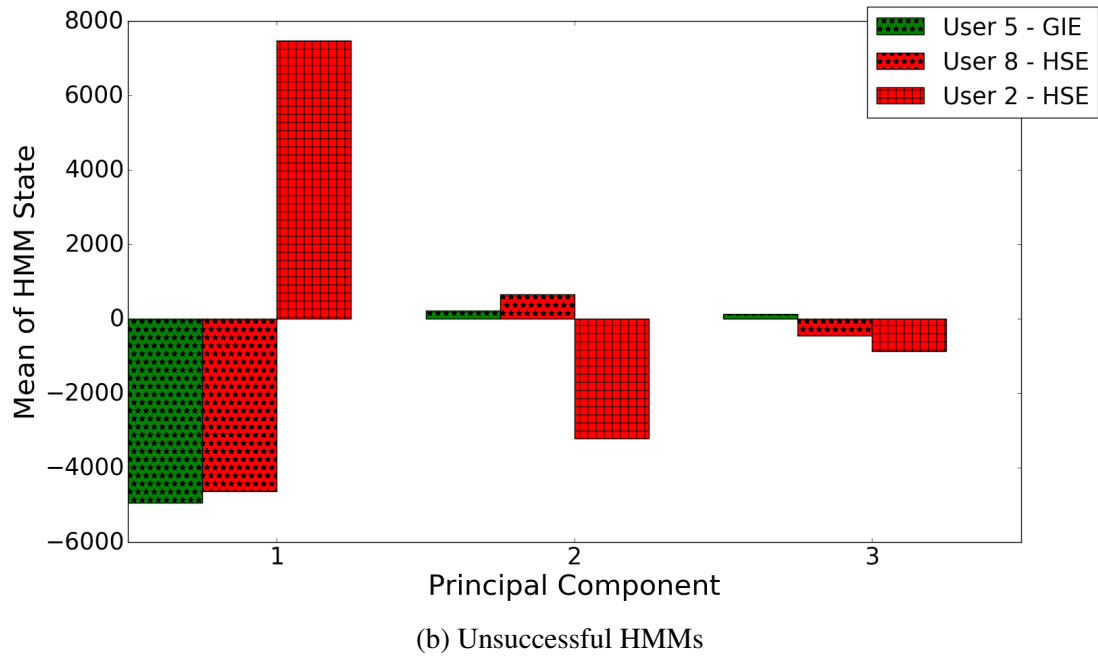
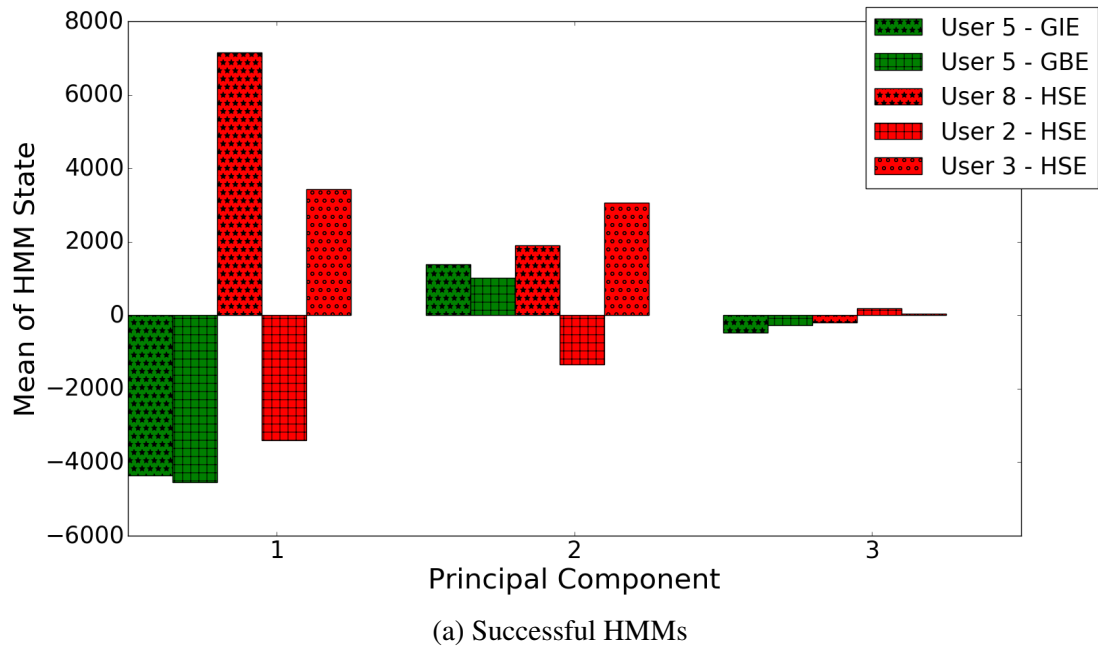


Figure 5.14: Displayed are the top three principal components of the final state of each user specific model for the object-action pair Drawer-Move. The values are separated in color by user models that performed well (green) and user models that performed poorly (red).

do differ, it is unclear if this difference is enough to account for the poor performance. As a result we do not show the rest of the object-action pairs. When we compare the variance of the means of the set of good user models and poor user models (in Figure 5.15) there is a clear difference in the variance of the HMMs that do well vs. poorly. Aside from Pastajar-Move, the poorly performing HMM models have a much larger variance in the observation state of the HMMs. The high performing models clearly had greater consistency in the observational values as opposed to those from the poorly trained models. This supports our hypothesis in Section 5.6.4 that the HMMs were having difficulty capturing a larger variety of demonstrations, whereas those trained with a smaller set converged to a specific and consistent observational state-space for the HMM. For Pastajar-Move, we believe the difference in variance is not high because none of the Pastajar-Move models themselves performed as well as the other object-action pairs.

5.6.6 Qualitative Observations

We presented qualitative observations from the user study based on anecdotes and common threads from a single open-ended question survey administered at the end of the user study in the previous results.

In general, users tended to view the hour long session as “fun” and compared getting the robot to successfully find the affordance to puzzle solving. For simple pairs like Pastajar-Move, users tended to get bored quickly and many wanted to move onto the next pair before the allotted time. The bread box was particularly favored because it was simple enough to provide many examples of success and failure, but “difficult” in comparison to the pasta jar.

Users’ dislike of failure resulted in an expressed preference to not provide examples of failure when teaching affordances. Not surprisingly, people dislike failure. However, it was surprising that users preferred not to provide example of failure even when instructed that providing negative examples of an affordance could be beneficial. Users were allowed to

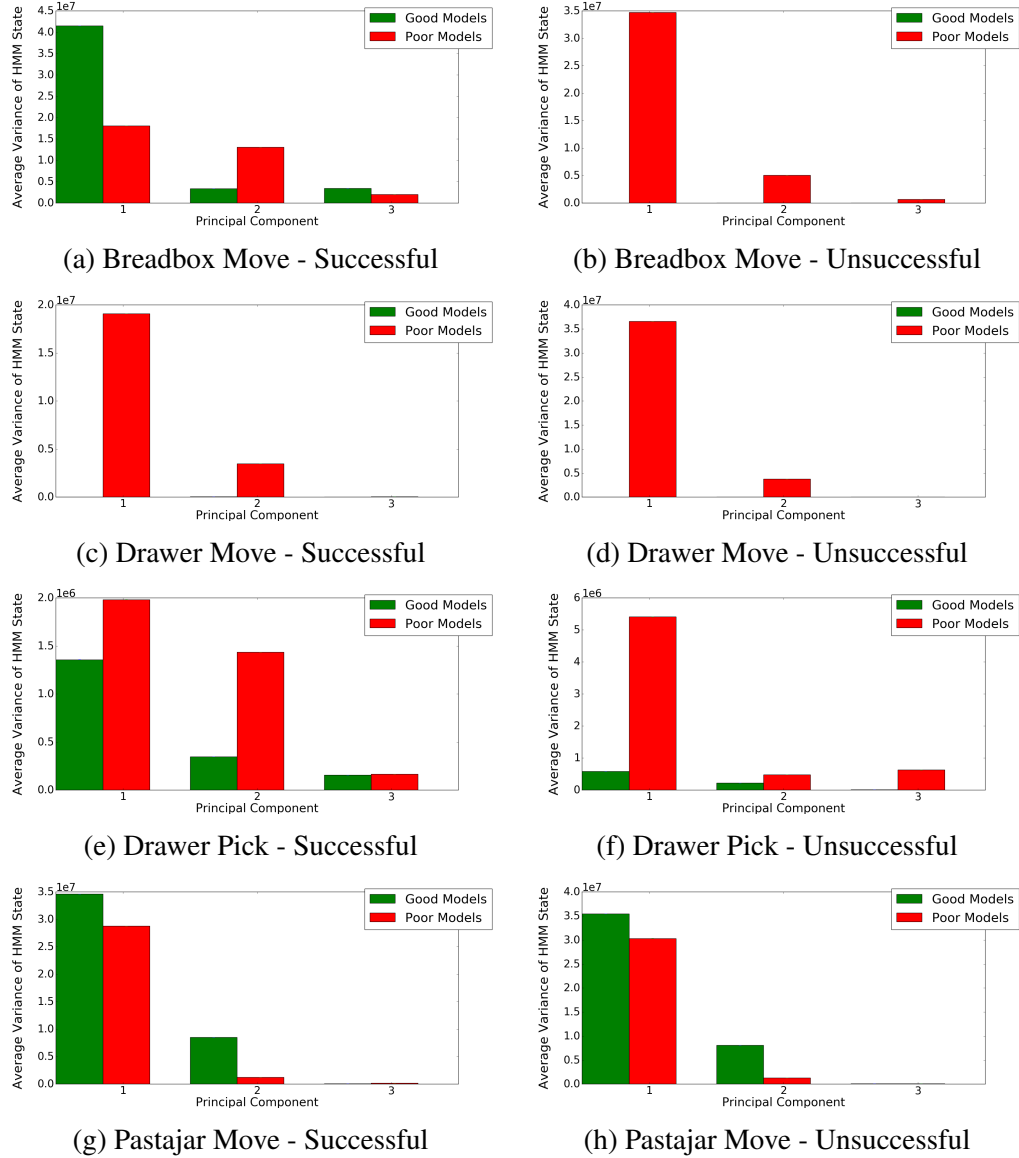


Figure 5.15: Shown are the comparisons of variance across the mean HMM values between good user models and poor performing user modes for all four object-action pairs. The means are separated by successful HMMs and unsuccessful HMMs. Poor performing user models overall have higher variance than high performing user models.

discard demonstrations and one user used this as a feature to “test the action [they] wanted to teach [the robot] without her recording to see if her interaction with the object would behave as [they] expected.” Another user reported feeling dejected that he could not get the robot to successfully find the affordance and felt that it was due to a lack of ability and intelligence. Interestingly, while only a few negative examples were provided, 6 out of 10 users reported thinking about providing negative examples in the survey.

Another common thread in the reported teaching strategy was the focus on providing “different ways to achieve the same outcome” and “show[ing] the affordance in multiple ways”. Half of the users reported using this method in their teaching strategy. This thread is interesting because it could account of the difference in variance across users when showing examples of interactions.

While users focused on providing varied and different interactions for the same effect, The vast majority of users did so by changing the robot’s action as opposed to the environment. Going into the study, we believed that users would take advantage of the fact that they could modify the environment as opposed to reteaching actions to provide different interactions. For example, showing a negative example of Pastajar-Move could be achieved but just putting the jar out of reach and this was in fact demonstrated to all participants before the study began as part of the tutorial on affordances. Even with this priming, users did not use this strategy, with only one user mentioning that they would “slowly modify the environment by repositioning the object”. For the users that did reuse an action, this generated very similar interactions since the same action is executed with a slightly different object position.

Finally, users displayed various failure recovery strategies. For example, turning on the lamp was a very difficult task due to compliance of the arm. As a result, some teachers experimented with several different grasps and directions of motion whereas some teachers quickly deemed the task impossible. This highlights two interesting components of LfD. First, it is important for users to *see* the result of their demonstrations as users quickly adapt

their teachings based on the robots actions. This is important to note because not all LfD techniques are evaluated in real-time due to the difficulty of robustly generating trajectories from demonstrations. The second is that not all teachers deal with failures similarly and a deeper understanding of user motivation is necessary to determine how to best motivate people to carefully teach robots a difficult skill.

5.7 Findings of Human-Guided Robot Self-Exploration

In this chapter, we provided an in-depth comparison of three different approaches to affordance learning: self-exploration, human-supervised exploration, and a *combined* human-guided approach defined as self-exploration biased by information provided from human teachers. Results showed that a combined approach, GIE, can learn affordance models on par with those generated from exhaustive SE, but using an order of magnitude fewer interactions with the object. The results of an individual analysis of each user-specific model provide several interesting pieces of insight that can guide future work for learning affordances from naïve users.

5.7.1 Variation in Teaching

To characterize the difference between users, we looked at clustering individuals based off of their demonstrations. However, while clustering the users based off of the EEF pose clearly showed that users provide very different approaches to teaching an affordance, merely clustering users based off of this was too simplistic to improve performance across all affordances. More importantly, we discovered that the impact of positive vs. negative interactions plays a large role in the performance of the users. We show that having many successful interactions causes the performance of individual models to decrease. Given that end-users do not intuitively provide many examples of failure, this suggests we need to explicitly ask users to provide more examples of failure.

While our hypothesis suggests that we should gather from users very similar successful

interactions to model the affordance with a low amount of data due to the impact of F/T sensing, this is misleading because what we really need to learn are all of the different ways it feels like to find the affordance. This suggests that not only do we need to gather varied interactions, but we also need to develop new modeling techniques that can capture the high variability due to F/T sensing (whether that be with a different representation where we model F/T felt with respect to the object or generate a library of models that encompasses this variance).

5.7.2 Action Generation

While the results and analysis of this chapter provide concrete guidelines for generating models from naïve users for object-action pair learning, there remains the open question on how we can use these models for the ultimate goal of task execution using affordances. Assuming a robot is given a task plan that requires a series of objects with specific affordances, the robot needs to locate objects in the room with the candidate affordances and test these objects to see if they can be used to perform the task. Currently, this system only addresses the first half of the equation where we are answering the question of “how can the robot learn about the object efficiently”? As we outlined in Section 3.5, the second half requires the robot to apply an existing learned model to a new object with a similar affordance and test the object for that affordance. For this to occur, the robot needs to generate actions from its existing models. While this chapter does not look into generation of trajectories from learned models, Chapter 6 and Chapter 7 will look at this specific problem of generating actions for the transfer of affordances.

CHAPTER 6

AFFORDANCE TRANSFER AND MULTISENSORY INPUT

In Chapter 3, we introduced a framework for affordances that relies on an agent performing an action on the environment to produce an effect. We defined how this creates a supervised learning problem between object-action pairs and effects. Specifically, we looked at how affordances could be learned and evaluated using binary classification. In Chapter 4 and Chapter 5, we demonstrated that object-action pairs could be used to monitor unseen interactions with various affordance and sensory inputs. We also demonstrated that non-expert human teachers could provide demonstrations that the robot could use to generate exploratory movements. The data collected using these actions produced object-action pairs that performed well at monitoring the success or failure of an unseen interaction.

In the final chapters of this thesis, we dive deeper into the insights gained from Chapter 4 and Chapter 5. In particular, we look at how robots can generate actions to transfer affordance models to adapt their actions to novel situations as well as how sensory modalities differ in nature and the impacts of those differences.

6.1 Affordance Transfer

By using the framework outlined in Section 3.5, we were able to explore a diverse range of objects and actions in Chapter 4 and Chapter 5. From this exploration, we start to gain insight on how high-level affordances (*e.g.* open-able) can be broken down into low-level affordances (*e.g.* lift-able)

6.1.1 Hierarchy of Affordance

As described earlier, affordances can be broken down into several components seen in Figure 3.1. The specific (object, action, effect) triple is often used in robotic affordance

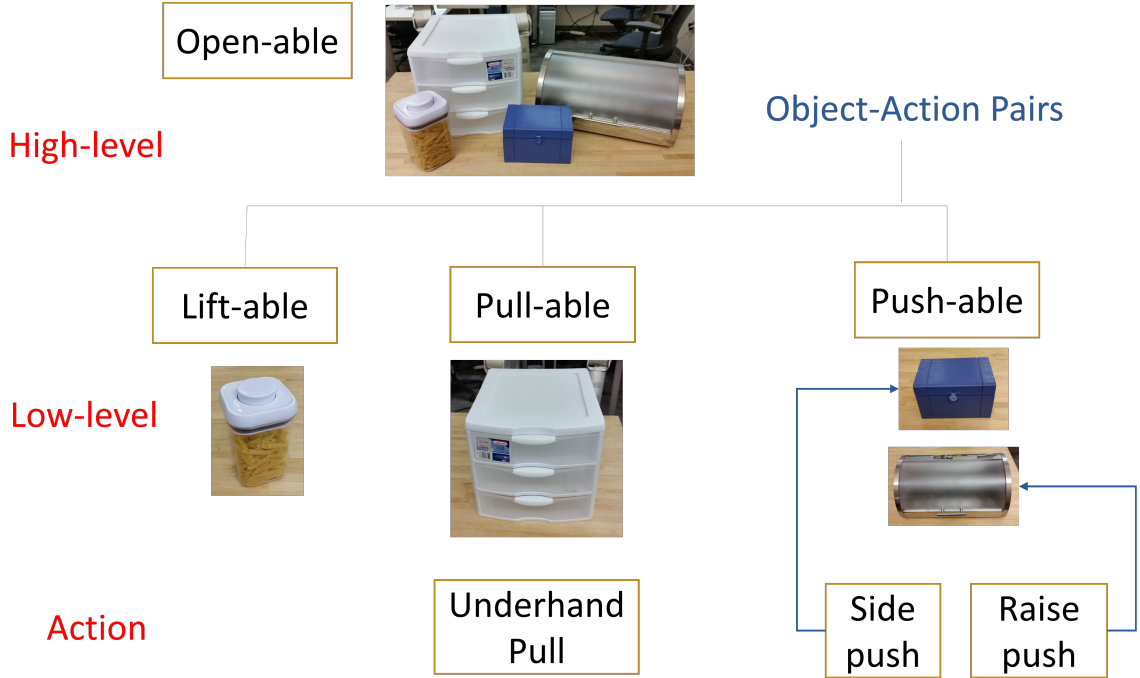


Figure 6.1: Affordances can be broken down into high-level and low-level affordances that relate directly to the actions and effects used to represent each affordances. High-level affordances can be broken down into several low-level affordances, which then have simple primitive actions that are directly related to the low-level affordance.

Table 6.1: Affordance Triple Subcategories

Category	Triple	High-level	Low-level
1	(same object, same action, same effect)	Same	Same
2	(different object , same action, same effect)	Same	Same
3	(different object, different action , same effect)	Same	Different
4	(same object, different action , same effect)	Same	Different

work (Şahin et al., 2007; Moldovan et al., 2012; Krüger et al., 2011). Within affordance transfer (where one affordance model is used on another object), the primary focus has been on detecting object components and features that indicate what subset of primitive actions should be used on the new object. However, these methods rely entirely on exploration and breakdown for tasks that require more dexterous manipulation. Instead, we look at the different components within the affordance triple and decide a transfer scheme based on what has changed within the triple.

To transfer affordances, it is important to understand how affordance models can be

associated with two “levels” of affordances. An example of this can be seen in Figure 6.1. On the “low-level”, a drawer that is opened by pulling on the handle has a “pull-able” affordance. However, on a “high-level”, a drawer is “open-able”. Understanding how these levels relate to the affordance triple gives insight on *how* to transfer an affordance. We can break down the transfer of affordance into several subcategories, which can be found in Table 6.1, that relate to “high-level” and “low-level” affordances. The subcategories do not contain all combinations of changes that can occur within the triple. In particular, we do not include any combinations where the effect changes because, in these situations, the affordance no longer remains the same and the transfer cannot be completed.

For example, let us examine the following combination (same object, same action, *different effect*): a robot pushes an object on the table with the result being that the object either slides or tips over. In the former case, the affordance is push-able. In the latter case, the affordance is tip-able. While it might be tempting to define all objects as their “low-level” affordance to make transfer easier, “high-level” affordances are critical to task planning. For example, for a robot to open a jar, the robot needs to understand open-able. If only the low-level affordance of lift-able were modeled, the robot would not be able to connect the task of opening the jar to the action of lifting the lid off of the jar.

6.1.2 Affordance Category Breakdown

By breaking down each of the individual categories by their triple, we can now look at how the categories differ and what role that plays for transfer:

- **Category 1 - same object, same action, same effect:** In the simplest case, an affordance model trained on a specific object should be able to handle perturbations of that object.

It is non-trivial to develop controllers that a robot can use to take into account the current state of the object. For example, if a drawer is already half-way open, the robot should only pull enough to fully open the drawer.

In this particular case, the robot uses the object state as feedback. However, it is important to note that the affordance action primitive remains the same (*i.e.* the robot still needs to pull on the drawer). The effect also remains the same. When all three triples are the same, both the “high-level” and “low-level” affordance remains the same (*e.g.* the mechanism for how to open the drawer is the same because the drawer has not changed).

- **Category 2 - *different object, same action, same effect*:** The next progression of affordance transfer looks at the case where an object has changed, but it still uses the same action primitive. For example, the two lamps (as shown in Figure 7.5) may be different from each other but the overall action primitive is the same. The robot still needs to pull down on the lamp chain to turn on the lamp.

Interestingly enough, even when the object has changed, both the “high-level” and “low-level” affordance model remains the same because we use the same primitive action for the new object (*e.g.* pulling the chain). This type of transfer has been studied for simple affordances (*e.g.* pushing, driving around) where controllers are often custom-generated for each affordance (Hermans et al., 2013a; Dogar et al., 2008).

- **Category 3 - *different object, different action, same effect*:** For this category, you can imagine having two different types of lamps: a push-button lamp and a pull-chain lamp. While both lamps have the same effect of increasing the light in the room, they require very different action primitives to turn on the lamp. Transferring between affordances becomes less straight-forward. Although the two lamps have the same high-level affordance of providing light, they have very different low-level affordances (*i.e.* push-able vs. pull-able).
- **Category 4 - *same object, different action, same effect*:** In the last category, we can use a breadbox as an example. Different actions can be taken to open the bread-

box, including pushing the lid or grasping the handle and pulling. Given that the underlying action has changed, the low-level affordance also changes. To discover this new action type would require giving the robot new knowledge (i.e. another demonstration) or allowing it to explore the object.

In this work, we assume the robot is given the category of the object-action pair. Future work could employ affordance knowledge bases (Zhu et al., 2014) and hierarchical task planning (Mohseni-Kabir, 2015) to determine the category of an object-action pair.

6.1.3 Transfer by Category

For Categories 1 and 2, the high-level and low-level affordances remain the same. This allows us to use the same action to find the effects of a particular high-level affordance. We transfer affordances in these two categories through adaptation. Specifically, we look at methods that allow us to adapt existing action controllers to different configurations and novel objects.

For Categories 3 and 4, we believe that transfer between two objects with differing “high-level” and “low-level” affordances should not be done with the same mechanism as transfer between the first two cases (Categories 1 and 2). In particular, these categories of transfer require additional knowledge of the object, such as the articulation mechanism. To address this additional constraint, we believe transfer in these categories requires an additional step prior to object adaption. Specifically, the robot should build a library of object-action pairs. During transfer, the robot could either select the most likely object-action pair or cycle through its existing library and perform adaptation. However, if none of the existing object-action pairs in its library work, the robot should perform additional exploration as seen in (Wang et al., 2014) or ask for additional guidance through demonstration (much like how this thesis learns new object-action pairs). While it is possible to perform transfer for Categories 1 and 2 with this approach, it is inefficient to perform exploration for these categories as we already know what low-level object-action pair to

perform.

6.2 Multisensory Feedback

In Chapter 4, we discovered that some haptic affordances were easier to model than others. One insight from that experiment was that object-action pairs that had a distinct continuous goal signal (*e.g.* the weight of the lid vs. no weight) performed better than those that were discrete (*e.g.* force change to push the lid of a box open). A wide variety of useful sensory feedback can be gathered when a robot performs different tasks. Furthermore, within each modality, feedback can be broken down into two categories: discrete and continuous.

Take, for example, the task of opening the drawer in Figure 4.4. The interaction of pulling open the drawer is continuous across all modalities. The robot continuously feels a force while pulling on the drawer, visually sees the drawer get “larger”, and hears the scraping of the drawer opening over time. Interestingly, the goal criteria for successfully opening the drawer is discrete in that there is a sudden increase of force and torque when the drawer cannot be opened any further and a “thump” sound when the drawer reaches the end of the rail. By contrast, not all modalities are continuous when you turn on the lamp in Figure 5.2h. Here, only the force felt while pulling on the chain is continuous whereas the sound of the lamp turning on and the sudden change in force are discrete. Visually, the light sends a discrete signal when it first turns on. Later, as it remains on, it becomes a continuous signal. Understanding these two categories of feedback provides insight on what goals are harder for robots to detect.

6.3 Contributions

In this thesis, we focus on how to perform adaptation of low-level affordances. In particular, we perform adaption for Categories 1 and 2. Furthermore, we demonstrate low-level affordance transfer with more complex tasks than prior work. It is important to note that this approach supplements existing work where additional exploration is used to transfer

affordances in Categories 3 and 4 (Wang et al., 2014).

Furthermore, we look at the differences between continuous and discrete sensory modalities. We show that a robot can improve its ability to detect success by combining both discrete and continuous signals. In the following chapter, we explore two different objects and the role that sensor modalities plays in adapting object-action pairs to the environment.

CHAPTER 7

MULTISENSORY AFFORDANCES FOR ADAPTIVE OBJECT MANIPULATION

As we outlined in both Chapter 3 and Chapter 6, the goal of learning affordances is to give the robot the ability to perform tasks in human environments. In Chapter 4 and Chapter 5, we focused on the situation where the robot finds the same object it learned the object-action pair on. In this chapter, we address the second challenge of how the robot can adapt to objects that it has not previously encountered. Furthermore, we show that multisensory inputs improve adaptation. Specifically, in this chapter, we address the questions of "*How can a robot learn adaptable controllers that use multiple sensory inputs, and demonstrations from human teachers?*" and "*What role do sensory inputs (visual, haptic, and auditory) play in affordance representation?*" outlined in Chapter 3.

In Chapter 6, we proposed the notion of categories for affordance transfer. In this chapter, we introduce Real-time Multisensory Affordance-based Control (RMAC). RMAC is a novel approach that enables robots to adapt existing object-action models using multisensory inputs for Category 1 and Category 2 affordances. This transfer is useful in situations where the robot has already learned an existing object-action pair but still has to modify existing controllers for slight changes in the environment. For example, a robot mixing batter may already have an object-action model for stirr-able with a spoon but how can the robot modify its controllers when stirring with a fork? Furthermore, what feedback (*i.e.* sensory channels) should the robot controllers use to determine when the batter is ready? To address this specific problem, we take a two-pronged approach where we (1) use the same framework outlined in Section 3.5 to model the robot's actions using affordances and (2) expand the aforementioned framework to represent affordances as a sequence of multisensory segments.

To build a robot controller that can adapt in these situations, we use segmentation to



Figure 7.1: Robot platform, Prentice, turning on a lamp

break down a robot’s trajectory into subskills (Niekum et al., 2015b; Konidaris et al., 2012; Kroemer et al., 2015; Righetti et al., 2014; Chebotar et al., 2014). These subskill segments represent components within a “low-level” affordance. By identifying these components of the “low-level” affordance, we isolate locations in the affordance where the robot can modify its trajectory and still successfully manipulate the object. Furthermore, we use a multisensory representation of the environment. For example, while a robot could rely on visual information to determine if a lamp similar to the one seen in Figure 7.1 is turned on, it can also utilize touch to detect the change in pressure and sound to hear the click of the switch. This allows the robot to naturally develop contingency cases (e.g. light bulb is broken). A robot could adapt its control schemes to the environment by using feedback on each of its sensor modalities (e.g. pull until it feels a particular force, hears a click, or sees light).

This example shows how manipulation skills can be represented as subskills (e.g. pull, grasp), which connect naturally to the different sensory modalities. Recent work shows that multiple sensory modalities can be modeled via data collected through robot exploration and improves robot manipulation (Sinapov et al., 2011b; Sinapov et al., 2014a), especially when it comes to detecting anomalies (Park et al., 2016) and performing failure

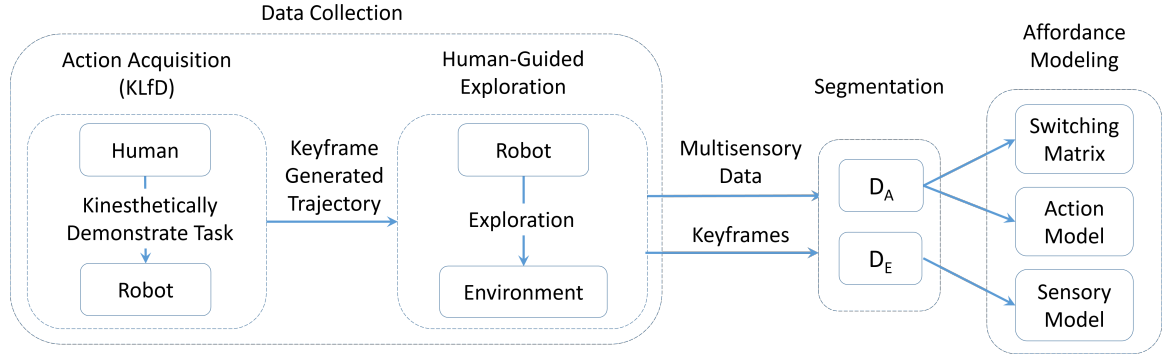


Figure 7.2: Real-time Multisensory Affordance-based Control (RMAC). The robot first collects multisensory data using human-guided exploration. The data is then broken into two sets of subskill segments that are used to generate a switching matrix, an action model, and a sensory model to represent the affordance.

recovery (Kappler et al., 2015).

7.1 Approach: Real-time Multisensory Affordance-based Control (RMAC)

To build a system that can adapt a learned affordance model to changes in the environment, we look to the hybrid control community where prior work (Mason, 1981; Raibert and Craig, 1981; Khatib, 1987) has shown that manipulation skills can be broken into segments that use different sensory spaces as feedback (*e.g.* position vs. force). Specifically, RMAC takes the following steps to achieve affordance-based control: (1) **Data collection:** We obtain a demonstration of the skill, and execute this demonstrated trajectory to collect positive and negative examples of achieving a particular object affordance. (2) **Segmentation:** We segment each trajectory into subskill segments. In this work, we use keyframes to help in the segmentation process, but this could be easily replaced with other segmentation algorithms (Niekum et al., 2015a; Kroemer et al., 2015). (3) **Affordance modeling:** For each segment, we build an action and sensory model. We represent the sensory model using left-to-right HMMs trained with multisensory positive data. This allows us to determine (a) over time, where we are progressing within a segment and (b) whether or not we have progressed to the end of the trajectory for the segment. (4) **Adaptive object manipulation:** We can recreate the affordance by using these learned subskill segments for object

manipulation. This section covers each aspect of RMAC in detail. An overview is seen in Figure 7.2.

7.1.1 Data Collection

We use the same method of data collection as described earlier in Chapter 3. For this chapter, we use human-guided exploration as described in Chapter 4, where the robot executes a trajectory generated using the provided keyframes while a person modifies the environment to collect varied interactions.

7.1.2 Segmentation

While recent work in trajectory learning for LfD have had success with automatic segmentation (Niekum et al., 2015b; Niekum et al., 2015a; Kroemer et al., 2015; Chu et al., 2017), many of the techniques require careful hand-tuning and are specific to the task selected. These methods are noisy enough that state-of-the-art manipulation systems still provide expert segmented trajectories (Kappler et al., 2015). In this work, we take a different approach and use keyframes (Akgun et al., 2012b) to segment trajectories. As described in Chapter 3, people are goal-oriented (Csibra, 2003; Meltzoff and Decety, 2003) and keyframes provide points in time where important parts of the trajectory are changing. Furthermore, the trajectories that the robot executes are generated from these keyframes. Therefore, not only are these snapshots important to the skill, but they are actually used in the parametrization of the skill. For example, to turn on the lamp seen in Figure 7.3, the keyframes provided are the start, the approach, the grasp point, the pulling point, the retract point, and finally the end. Figure 7.7 shows that the changes in the sensory space correlate well to the keyframe changes. We refer to each of these segments as subskill segments.

In RMAC, we generate two sets of subskill segments. The first set (D_A) represents the actual location the trajectory should be split and the second (D_E) represents extended segments that are slightly longer than the actual segmentation location. This is done to capture

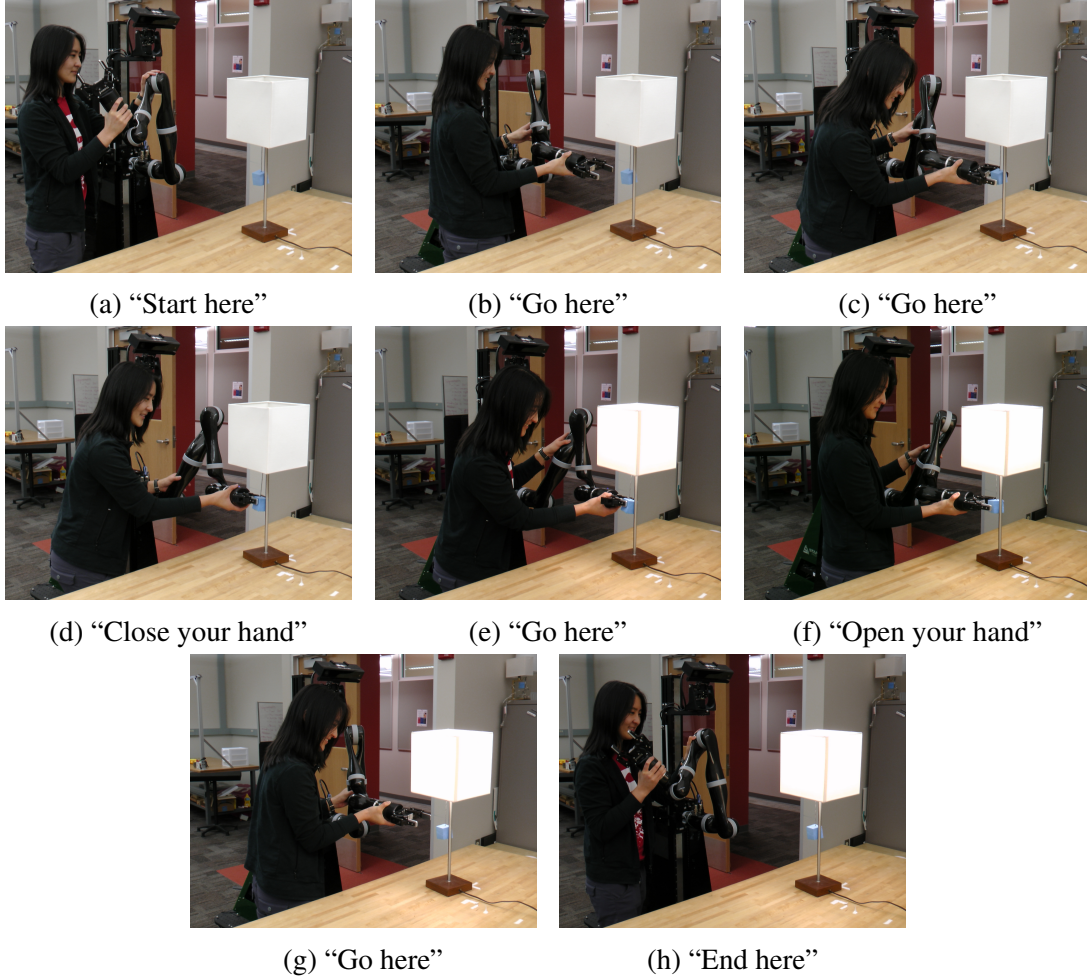


Figure 7.3: Kinesthetic demonstration using keyframes for turning on the lamp.

the sensory input of what the robot should expect when it has successfully completed the current subskill segment and needs to transition to the next. In this work, we generate D_A from the exact keyframe locations and D_E by going 0.5 seconds past the exact keyframe location. As mentioned previously, we use keyframes to segment in this work, but the specific segmentation algorithm (*e.g.* CHAMP, STARHMM, etc.) can easily be replaced. RMAC does not inherently depend on keyframe based segmentation, it only requires that actual and extended segments be provided to the algorithm. It is also important to note that while this work uses simple keyframes, it could also be extended to use hybrid keyframes, which inherently captures the velocity of a trajectory for tasks where the dynamics of the action matter.

7.1.3 Affordance Switching Matrix

Once we have identified each of the subskill segments, we generate a “switching matrix” for each skill that represents the high-level action (*i.e.* control mode) that the robot uses to move, similar to those in traditional hybrid control methods (Raibert and Craig, 1981). In this work, we have two control modes: pose and sensory. When in pose mode, the robot focuses only on how to get to the pose at the end of the subskill segment and does trajectory planning using inverse kinematics and end-effector pose. During sensory control mode, the robot’s movements depend on the direct feedback of the various sensory inputs the robot is receiving in real-time.

For each object-action pair, we represent the change in control modes as a single $M \times N$ matrix (S) where M represents the number of modes in the controller and N the number of segments. We can call S a “switching” matrix (a term colloquially used in the traditional hybrid control community) because S represents the transitional matrix that determines what modalities should be constrained and what degrees should be free to move for each segment. For this work, M is two (pose and sensory modes). Traditionally, M represents the different constraints in Cartesian and sensory space. In this work, we merge the indi-

vidual constraints into a single value and assume all Cartesian directions or sensory inputs matter (*i.e.* the exact vector (x,y,z) is important to each subskill segment) as opposed to saying only a specific direction is crucial (*e.g.* only z matters for an EEF to maintain constant contact with a table). While RMAC could still be used without this simplification, we do not explicitly state each constraint in the switching matrix and instead allow the model to capture the importance of each direction/modality. An example of S can be found in Equation 7.1.

$$S = \begin{bmatrix} pose(1) & pose(2) & \cdots & pose(n) \\ sensory(1) & sensory(2) & \cdots & sensory(n) \end{bmatrix} \quad (7.1)$$

Each column of S represents a subskill segment within the trajectory. Each row of S represents the control mode (*e.g.* pose vs. sensory). For each mode and for each segment we assign a binary value (0 or 1) to represent if that channel is constrained during that segment. For example, if the pose row of the S matrix seen above were $[1, 1, 0]$, the controller would utilize pose control for the first two segments and the third segment would utilize sensory control.

Although we use an expert to hand label which control mode the robot should use for each subskill segment, a switching matrix could be learned automatically by computing the variance through each subskill segment. Future work will explore the extent that this can be given by human end-users that are not experts with robots. Furthermore, similar to prior work (Kappler et al., 2015), we assume that the sequence of subskill segments that will be executed are pre-defined and the system will either naturally progress through each subskill segments, or stop if something has occurred that cannot be adapted.

7.1.4 Subskill Segment Modeling

Once we have each subskill segment, we create an action model and an effect model of each subskill. Prior work in this area typically represents the action model of a subskill segment

using DMPs (Kroemer et al., 2015; Niekum et al., 2015b; Kappler et al., 2015). These works then learn a high-level policy that dictates what DMPs to execute based on sensory models. These sensory models typically either utilized time and use HMMs (Kroemer et al., 2015) or discretize the effect space and use Support Vector Machines (SVMs) (Kappler et al., 2015; Su et al., 2016). While DMPs can adapt to slight perturbations in the scene, the goal (*i.e.* end-effector position) must be clearly defined. To address this issue, Kappler et al. (2015) build a library of DMPs and select the correct DMP to execute at every moment in time by tracking where the robot trajectory is in the DMP and comparing it to the expected sensory traces. While DMPs can track where along the trajectory the robot is executing using a phase variable, ρ , Kapper et. al show that it cannot be reliably used due to noise and perturbations in the environment. Instead, they track the most likely state of the current subskill segment by using a naïve Bayes classifier and SVMs for each state to indicate if the sensory traces are as expected. If the traces differ too greatly, the system selects a new DMP from the library.

In this work, we take a different approach and represent the action model of each subskill segment as a velocity vector v_n , where T is the total number of time steps in the trajectory q^n and q_t^n is the pose of the trajectory for subskill segment n at time t . v_n is generated from the the set of segments D_A .

$$v_n = \frac{1}{T} \sum_{t=1}^T q_t^n - q_{t-1}^n \quad (7.2)$$

This allows us to view each subskill segment in even smaller time steps than that of a subskill. For example, a subskill segment that has a robot pulling on a handle can be viewed as a sequence of small incremental steps away from the handle until a specific change in state is detected (*i.e.* a large force is felt). Representing the trajectory in the smallest possible step size increases the adaptability of the motion to changes in the environment. In particular, we no longer specify in task space where the robot must go, but instead rely on the effect space to determine if the robot has succeeded. Representing the trajectory

in the smallest possible time step poses an additional challenge that DMPs avoid: While DMPs give a clear ending position to the robot, RMAC requires the robot to determine when to stop.

To model the sensory space, we take a similar approach to that of Park et al. (2016) and model the sensory space using left-to-right HMMs. We train the HMMs using the segments (D_E). By modeling the sensor space with HMMs, we track both the most likely state within the subskill segment the robot is in as well as model the likelihood of experiencing the different sensory inputs in each state. Furthermore, we utilized time in the model whereas work from (Kappler et al., 2015; Su et al., 2016) do not because they rely on SVMs. To address the challenge of determining when the robot has finished a particular subskill segment, we track the current state of the left-to-right HMM. If the robot reaches the final state of the HMM, we determine that the robot has completed this subskill segment. While not in the scope of this work, these HMMs also allow us to determine when the robot has failed by tracking the likelihoods of an anomaly similar to that of (Park et al., 2016). Once the robot detects that it has completed this subsegment, it moves directly to the next segment. Although we specify the exact sequence of segments, this could easily be replaced with a high-level policy similar to ones in (Kroemer et al., 2015; Akgun and Thomaz, 2016).

7.1.5 Execution

Once we have built a switching matrix, action models, and sensory models for each subskill segment, we can now control the robot to adaptively interact with objects. Specifically, during execution, the following steps occur:

1. For each switching matrix, segments that have pose control perform an extra step where the user gives the segment a specific pose that the EEF must reach.
2. For non-adaptive segments, the robot computes a transformed trajectory relative to

the object and uses planning to get the EEf to the correct pose. Once the pose is executed, the segment’s HMM determines if we are ready to go to the next segment.

3. For adaptive segments, we execute the velocity, v_n , and collect the sensory feedback at each time step. After each step, we stay in the current segment unless we have encountered the final state of the left-to-right HMM.

7.2 Offline Validation: Adapting Learned Affordance Models to Changed Objects and New Objects

To evaluate the ability for the robot to adapt a learned affordance model, we choose two situations that vary in difficulty. The first looks at how to adapt to changes to a previously learned object while the second looks at how to transfer a previously learned model to a different object. Specifically, this looks at the two cases described in Section 6.1.2: Case 1 - same object, same action, same effect and Case 2 - different object, same action, same effect. Furthermore, for each object, we look at the role the different modalities play in adaptation and show that RMAC performs better with multisensory input. For this chapter, we use the “Prentice” robot.

7.2.1 Experimental Setup

For Case 1, we use the same drawer as the one used in Chapter 5. As seen in Figure 7.4, the drawer has 5 different configurations that are used for testing. Specifically, we have the robot try to open the drawer with the drawer already open at 2 inch intervals (i.e. 1in, 3in, 5in, 7in). We chose this as the first test of RMAC for several reasons. First, to transfer a learned affordance, there are different stages of transfer that include changes to the current object. By selecting a drawer, we can systematically determine how the controller can adapt to environment changes. Furthermore, we chose this drawer because it is light enough for the robot to drag across the table in the event that the robot does not stop at the correct location. Allowing the drawer to slide across the entire table prevents the robot

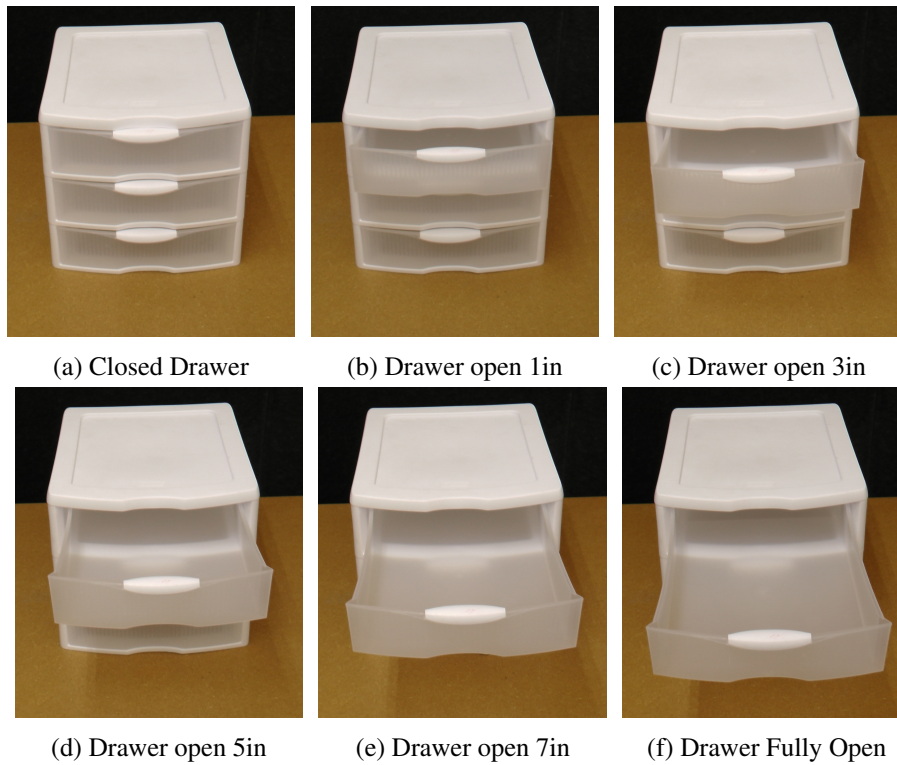
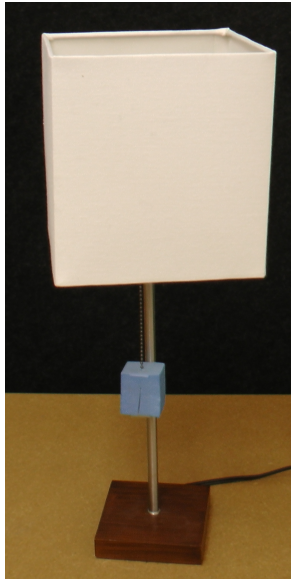
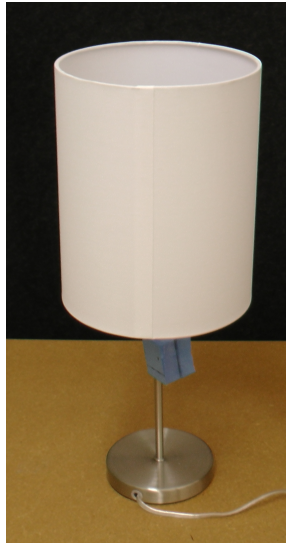


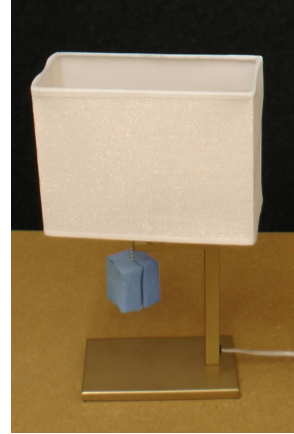
Figure 7.4: The various configurations the drawer is placed in for the robot to adapt its affordance controller. The robot pulls the drawer to fully opened from the 5 different starting configurations that change in 2 inch increments.



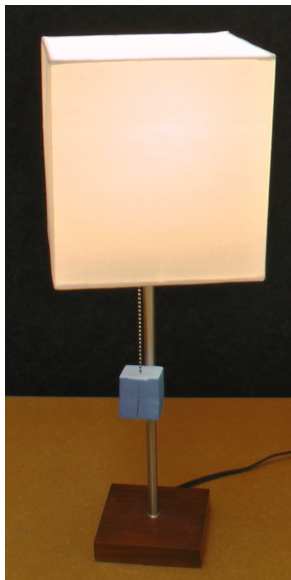
(a) Original Lamp - off



(b) New Lamp 1 - off



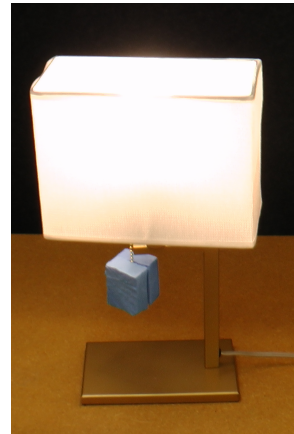
(c) New Lamp 2 - off



(d) Original Lamp - on



(e) New Lamp 1 - on



(f) New Lamp 2 - on

Figure 7.5: The various lamps the robot turns on in this chapter

from injuring itself if it continues to pull on the drawer past the fully open position. We can use the distance the robot drags the drawer as a metric to evaluate RMAC.

For Case 2, we use 3 different lamps with varying pull chains as seen in Figure 7.5 with the light on and off. We chose these objects to demonstrate the transfer of a learned affordance on one object to another object. In particular, we are interested in transfer of an existing affordance model to a novel object without requiring the robot to re-explore the object - a subject covered by prior work in affordance transfer (Wang et al., 2014). We focus on whether the robot can transfer the knowledge of the effects that it is seeking (*e.g.* light change, forces felt, etc.) to another object that also has these effects. Similar to the drawer, lamps were selected to minimize potential damage if the robot pulled too long.

7.2.2 Data Collection

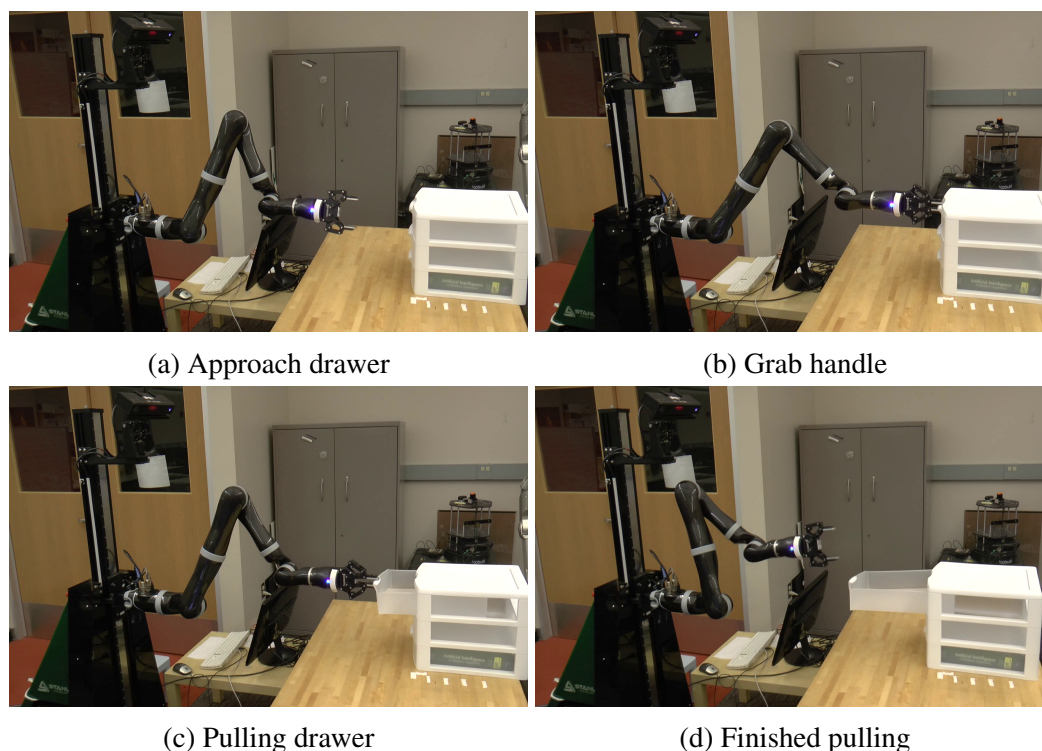


Figure 7.6: Prentice executing demonstrated trajectory on the drawer

As described in Chapter 3, we use a keyframe-based LfD approach (Akgun et al., 2012a). In this chapter, to replay the demonstrations, the keyframes are not splined to-

gether. Instead, a path is planned through the keyframes using Rapidly-exploring Random Trees (RRTs) (Kuffner and LaValle, 2000) after the end-effector pose is converted into joint space using TRAC-IK (Beeson and Ames, 2015). Prentice executes the trajectory autonomously on the object during playback as seen in Figure 7.6. This guarantees no external forces or torques are felt during data recording. We collected 50 interactions of opening a fully-closed drawer (Figure 7.4a) and 50 interactions of turning on a single lamp (Figure 7.5a).

7.2.3 Multisensory Features

Table 7.1: Sensor Data

Sensor	Modality	Data	Resolution	Features
JACO2	Haptic	GC Force	100 Hz	Raw Forces (F_x, F_y, F_z)
JACO2	Haptic	GC Torque	100 Hz	Raw Torques (T_x, T_y, T_z)
Robotiq	Haptic	Position	100 Hz	Raw Position (G_p)
Kinect2	Audio	Sound	44.1 kHz	Power/Energy (A_e)
Kinect2	Visual	Point Cloud	7 Hz	Color (V_{RGBA}), Volume (V_{vol})

We collect data from the sources shown in Table 7.1. From each data source we compute several features that are used to train the sensory model. For touch data, the robot collects the gravity compensated F/Ts at the end-effector and the gripper position. The haptic data is left as is, without any data preprocessing and the resulting features are the raw forces (F_x, F_y, F_z), raw torques (T_x, T_y, T_z) and raw gripper position (G_p). The raw audio data is recorded at 44.1 kHz. We compute root-mean-square (RMS) of the energy of the Short-time Fourier Transform (STFT) of the audio signal. The specific parameters used to generate the feature (A_e) are frame length: 2048 and hop window: 512. We use the python audio library librosa (McFee et al., 2015) to compute the audio feature. For the visual input, we compute two different features from the point clouds: the RGBA (V_{RGBA}) and the volume of the object (V_{vol}). We use the segmentation method from Trevor et al. (2013) to segment the object from the table after the plane is found using RANSAC (Fischler and

Bolles, 1981). To align the different data sources, we up- or down-sample the data to 100 *Hz*.

Figure 7.7 and Figure 7.8 shows the data for the computed features from each of the sensory channels. It also displays vertical lines for the location of the KF-segmented version of the trajectory. For the lamp in Figure 7.7, the frames can be viewed semantically as: (1) untuck the arm (2) approach the chain (3) close the gripper (4) pull down on the chain (5) open the gripper (6) back away from the lamp (7) retuck arm. For the drawer in Figure 7.8, the frames can be viewed as (1) untuck arm (2) approach the drawer (3) close the gripper (4) pull on the drawer (5) open the gripper (6) back away from the drawer (6) retuck arm.

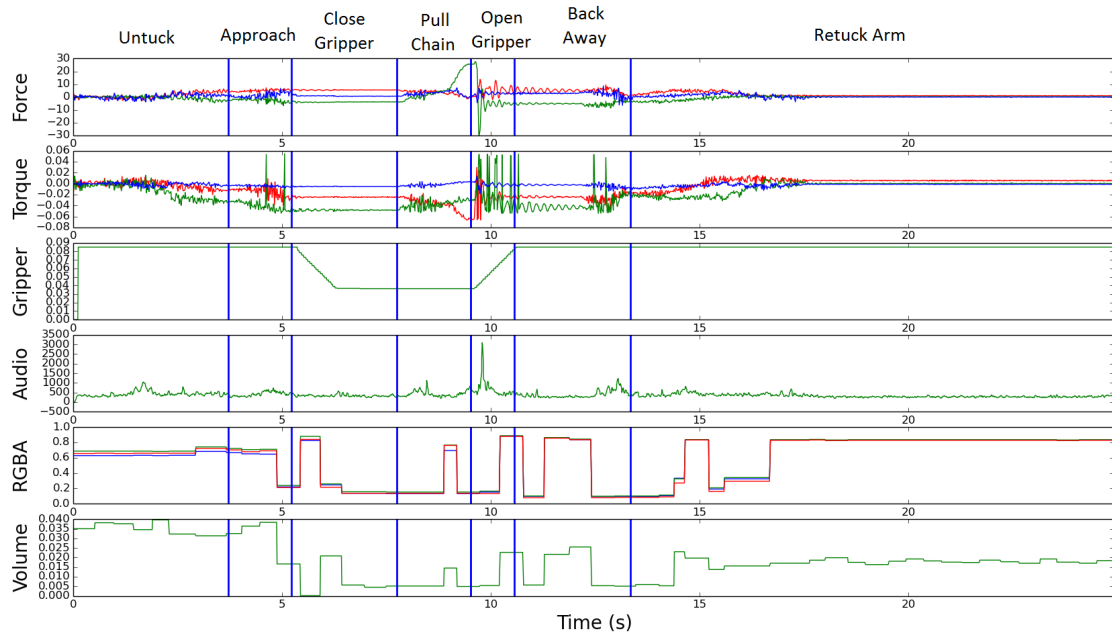


Figure 7.7: Computed features from one interaction with the lamp. The different sensory channels are displayed with vertical lines that indicate the location of the segments of the subskill segment set D_A

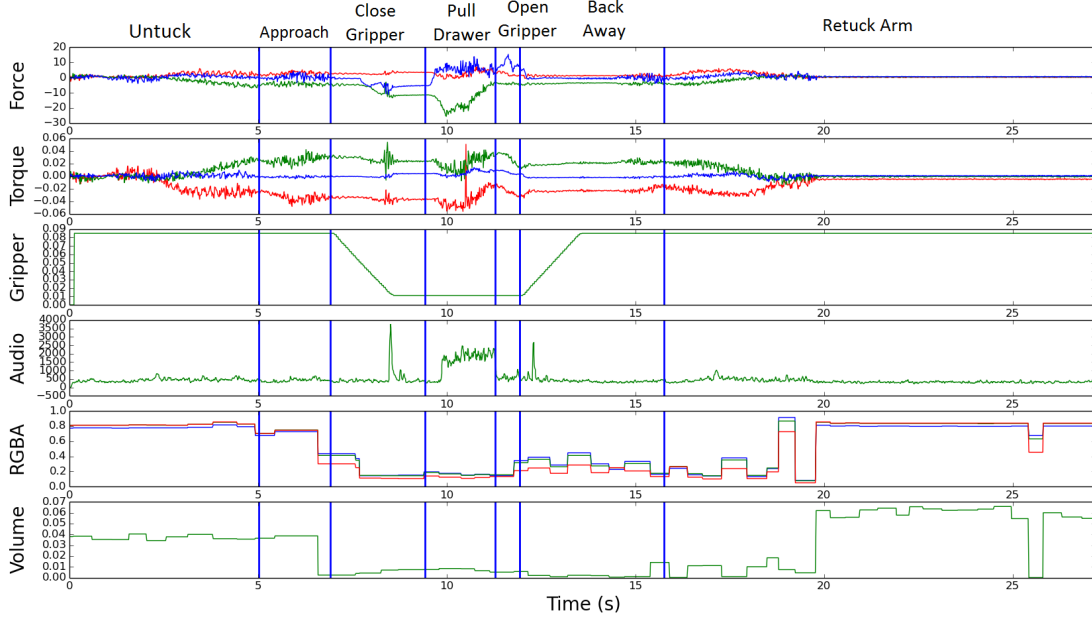


Figure 7.8: Computed features from one interaction with the drawer. The different sensory channels are displayed with vertical lines that indicate the location of the segments of the subskill segment set D_A

7.2.4 Training Sensory Models

As described in Section 7.1, we build an action and sensory model from the demonstrations. To test the importance of each sensory modality, we build 7 different sensory models for every combination of the three different sensory inputs (*i.e.* visual, haptic, audio). Specifically, we change the observation space O for the left-to-right HMM. The different combinations and feature spaces for O can be found in Table 7.2. Each observation space is modeled with a continuous multivariate Gaussian distribution.

Table 7.2: Sensory Model Combinations

Combination	Observation Space
Haptic	$F_x, F_y, F_z, T_x, T_y, T_z, G_p$
Visual	V_{RGBA}, V_{vol}
Audio	A_e
Haptic, Visual	$F_x, F_y, F_z, T_x, T_y, T_z, G_p, V_{RGBA}, V_{vol}$
Haptic, Audio	$F_x, F_y, F_z, T_x, T_y, T_z, G_p, A_e$
Visual, Audio	V_{RGBA}, V_{vol}, A_e
Haptic, Visual, Audio	$F_x, F_y, F_z, T_x, T_y, T_z, G_p, V_{RGBA}, V_{vol}, A_e$

To train each HMM, we used the successful interactions from the collected runs (lamp: 29, drawer: 32) for each object (*i.e.* drawer and lamp). We select the best number of states (between 2-15 states inclusive) for the HMMs by performing 5-fold cross validation. To score the HMMs, we do not use the log-likelihood of the HMM, but instead use the distance away from the true segment switching point. The smaller the value (*i.e.* closer to stopping at the correct location), the better the score. We normalize each observation space by removing the mean and scaling the features to have unit variance. Note that we tested the trained models on a test set that was different from the training set.

7.2.5 Test Set

To systematically compare RMAC using different sensory inputs, we first collect a test set that can be used to evaluate each trained HMM. In particular, this test set has to be different from the training set. The training set collected contains data of the robot successfully finishing the particular skill. However, to fully test how well RMAC adapts to the changing environment, the robot needs to continue to execute its motion past the point that it should stop. Then, we can determine if the RMAC stops at the correct point in time or chose an incorrect stopping location.

To collect this test set, we do not collect data using the demonstration as we do in Section 7.1.1. Instead, we use the real-time execution controller described in Section 7.1.5 to collect the test set. This allows us to collect data that (a) simulates what the robot will actually experience when performing execution online and (b) allows us to collect the data past the actual stopping point (*i.e.* when the drawer is fully pulled out or the lamp has turned on). We modify the real-time execution during a sensory feedback subskill segment to ignore any sensory feedback and keep executing its velocity vector v_n to a specific stopping point significantly past when the robot should have detected the subskill segment has succeeded.

For each object and configuration, we collect 5 test interactions. Specifically, we collect

5 test runs for each drawer configuration and 5 test runs for each lamp. This results in 35 test runs for the drawer object across the 7 drawer configurations and 15 lamp test runs across the 3 different lamps. With these test interactions, we can compare the importance of each modality in adaptation. Specifically, the goal is to determine if RMAC can automatically select what modalities to focus on when given all sensory inputs without requiring an expert to provide this beforehand. Furthermore, by comparing all combinations of modalities, we can examine what modalities seem to be most informative to the task.

7.2.6 Results - Case 1: Drawer

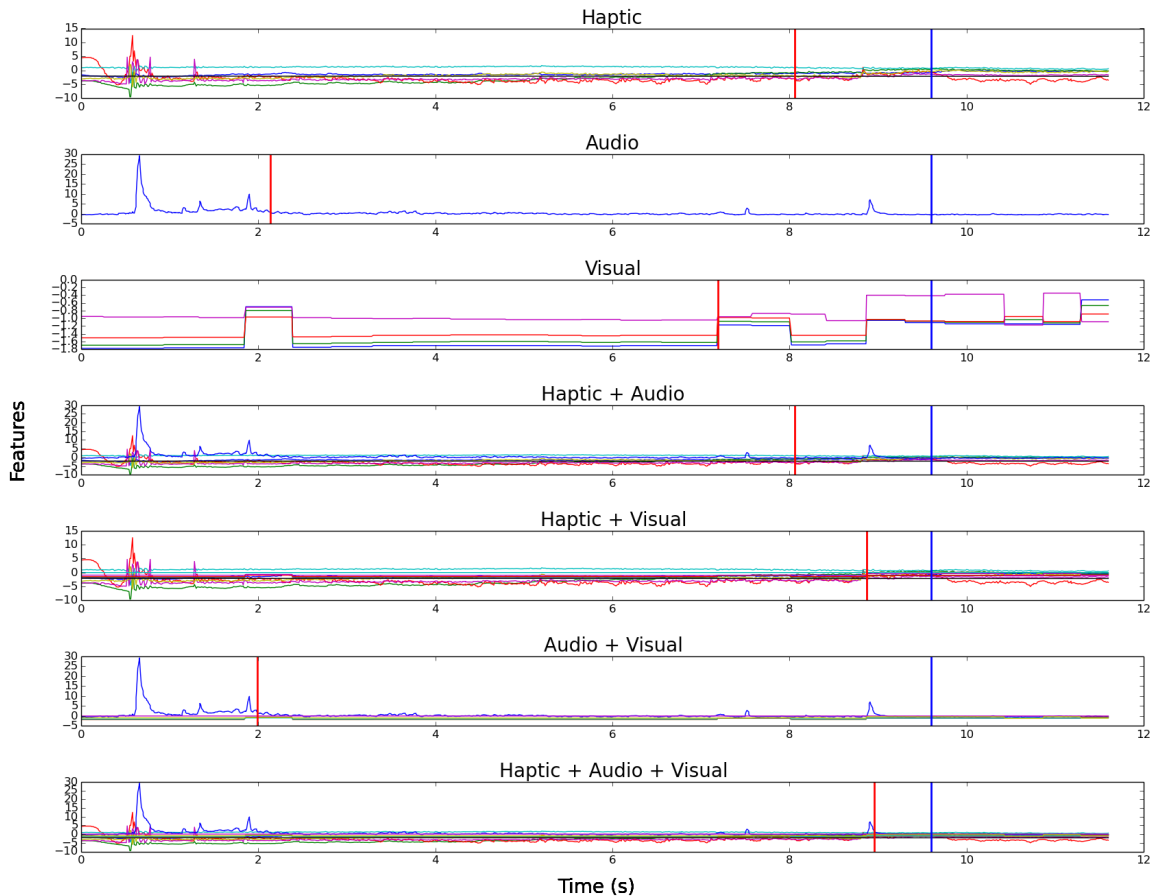


Figure 7.9: Opening the drawer comparison across the different modalities when the drawer is fully closed. The vertical red bars indicate where the model predicted it should stop and the blue vertical bars indicate where the ground truth stopping point is.

We can compare the result of selecting the correct stopping point using the different

sensory inputs in Figure 7.9. This figure shows one test run for the drawer configuration where the drawer is completely closed. Each subplot of the figure contains the data source (e.g. haptic, haptic and visual, etc.) as well as two vertical bars of differing colors (red and blue). The blue vertical bar indicates where the robot should have stopped. This value is hand-labeled by the author. The red vertical bar is where the real-time controller would have stopped using RMAC. The closer the red vertical bar is to the blue bar, the more accurate the controller is at stopping when the drawer is open.

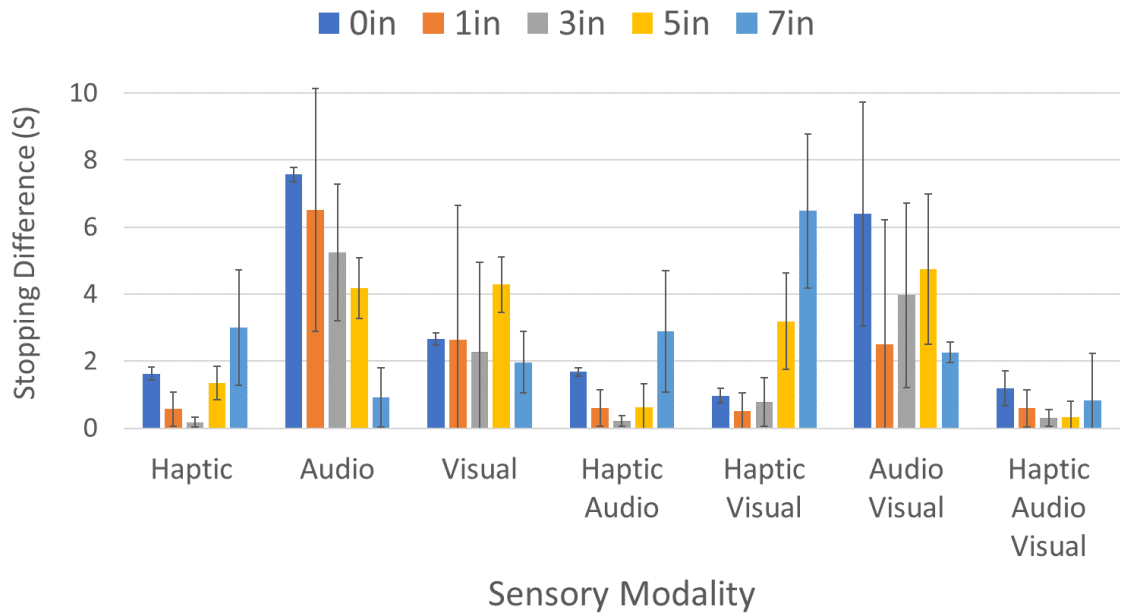


Figure 7.10: Average and standard deviation of absolute difference in stopping times for different drawer configurations across the 7 modalities.

Table 7.3: Drawer Fully Closed Detailed Times

Combo	1	2	3	4	5	Average
Haptic	1.54	1.61	1.38	1.92	1.71	1.63 ± 0.20
Audio	7.46	7.43	7.40	7.90	7.68	7.57 ± 0.21
Visual	2.41	2.53	2.66	2.85	2.83	2.66 ± 0.19
Haptic Audio	1.54	1.61	1.64	1.74	1.87	1.68 ± 0.13
Haptic Visual	0.73	1.08	1.00	0.78	1.25	0.97 ± 0.22
Audio Visual	7.61	9.59	9.12	2.82	2.81	6.39 ± 3.35
Haptic Audio Visual	0.64	1.57	1.38	0.65	1.71	1.19 ± 0.51

Table 7.4: Drawer Opened 1 inch Detailed Times

Combo	1	2	3	4	5	Average
Haptic	0.33	0.34	0.47	0.25	1.48	0.57 ± 0.51
Audio	7.73	8.41	8.52	7.88	0.06	6.52 ± 3.62
Visual	2.57	0.17	0.19	0.67	9.59	2.64 ± 4.01
Haptic Audio	0.33	0.34	0.25	0.51	1.56	0.60 ± 0.55
Haptic Visual	0.31	0.17	0.27	0.25	1.48	0.50 ± 0.55
Audio Visual	2.56	0.14	0.20	0.68	8.90	2.50 ± 3.71
Haptic Audio Visual	0.33	0.34	0.47	0.24	1.56	0.59 ± 0.55

Table 7.5: Drawer Opened 3 inch Detailed Times

Combo	1	2	3	4	5	Average
Haptic	0.03	0.25	0.08	0.14	0.42	0.18 ± 0.16
Audio	4.09	6.56	6.71	2.18	6.66	5.24 ± 2.04
Visual	1.21	3.04	0.12	6.62	0.36	2.27 ± 2.69
Haptic Audio	0.09	0.25	0.08	0.20	0.46	0.22 ± 0.15
Haptic Visual	0.08	0.79	0.60	1.99	0.42	0.78 ± 0.72
Audio Visual	5.96	5.92	0.11	1.98	5.86	3.97 ± 2.75
Haptic Audio Visual	0.08	0.25	0.08	0.66	0.46	0.31 ± 0.25

Table 7.6: Drawer Opened 5 inch Detailed Times

Combo	1	2	3	4	5	Average
Haptic	1.87	1.74	0.83	0.82	1.48	1.35 ± 0.50
Audio	3.66	4.80	2.81	4.94	4.67	4.18 ± 0.92
Visual	4.14	4.80	2.92	4.94	4.67	4.29 ± 0.83
Haptic Audio	1.68	0.05	0.39	1.00	0.04	0.63 ± 0.70
Haptic Visual	4.14	4.80	1.27	3.54	2.21	3.19 ± 1.44
Audio Visual	4.14	7.58	5.63	4.94	1.44	4.75 ± 2.24
Haptic Audio Visual	0.13	0.04	1.16	0.20	0.06	0.32 ± 0.47

Table 7.7: Drawer Opened 7 inch Detailed Times

Combo	1	2	3	4	5	Average
Haptic	4.46	0.01	3.56	3.64	3.31	3.00 ± 1.72
Audio	0.33	0.49	2.49	0.56	0.71	0.92 ± 0.90
Visual	0.35	2.25	2.49	2.27	2.51	1.97 ± 0.92
Haptic Audio	4.26	0.03	4.62	2.92	2.63	2.89 ± 1.81
Haptic Visual	8.34	8.07	7.94	4.76	3.31	6.48 ± 2.30
Audio Visual	1.76	2.25	2.49	2.27	2.51	2.26 ± 0.30
Haptic Audio Visual	0.10	0.01	3.33	0.52	0.21	0.83 ± 1.41

For this particular test run, audio information is not useful at all in determining whether the robot has opened the drawer. While visual information is helpful, it is not as informative as haptic information. Furthermore, the best combination of sensory inputs is that of visual and haptic feedback, which intuitively makes sense when we think about what we pay attention to when opening drawers. Finally, we can see that the combination of haptic, visual, and audio data does not perform worse than that of haptic and visual. This shows that the algorithm can determine what modalities matter automatically, without any external sources indicating which modalities are important.

To determine how the robot performs overall, we can look at Table 7.3, where the absolute difference in time between the ground truth stopping point and estimated stopping point is computed for each of the 5 test runs across all 7 modality combinations for the configuration where the drawer is completely closed. Since we want the estimated point to be as close to the ground truth stopping point, the smaller the distance away in time, the better the score. Overall, we can see the same trend across all test runs: the combination of haptic and visual performs best while merging all modalities results in similar, though slightly lower, performance.

This trend can be seen across all five configurations in Table 7.4, Table 7.5, Table 7.6, and Table 7.7. We can also see this in Figure 7.10 where the average distance away in time to ground truth are shown for all five configurations. It is interesting to note that while the configuration of having the drawer 7 inches open is the most difficult to detect because there is only a short amount of time the robot is pulling before it stops, it does equally as well as when the drawer is fully closed. Furthermore, these results show that the robot can adapt to different configurations and, importantly, this adaptation occurs without the robot ever being trained on any configurations other than fully closed.

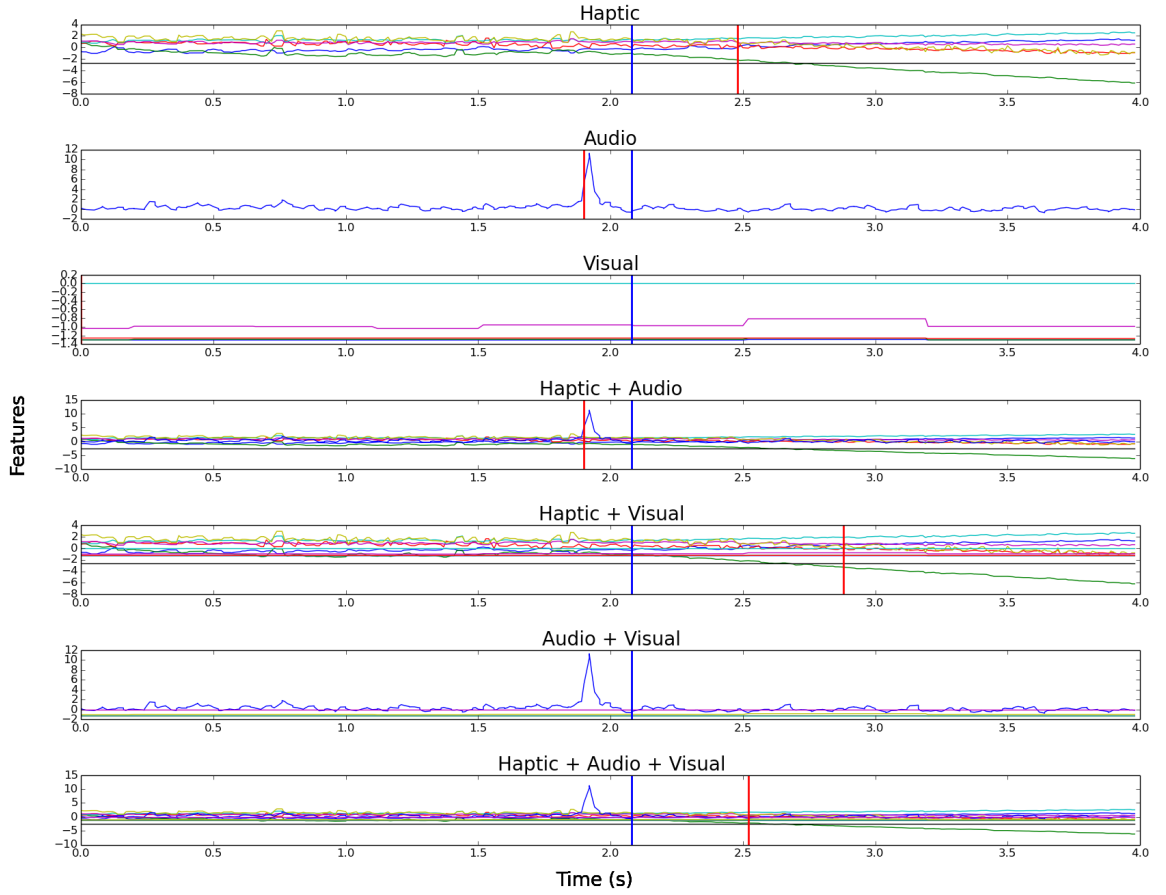


Figure 7.11: Comparing turning on the New Lamp 2 across the different modalities. The vertical red bars indicate where the model predicted it should stop and the blue vertical bars indicate where the ground truth stopping point is. For *Haptic*, *Haptic+Visual*, *Haptic+Audio+Visual*, the robot stops after turning on the lamp. For *Audio + Visual* the robot never stops pulling. For *Audio*, *Visual*, *Haptic+Audio* the robot stops too early.

7.2.7 Results - Case 2: Lamps

For this second case, we compare a more difficult situation where the object has changed. While the interactions with the drawer occurred with the particular drawer, the lamp requires that the object-action model be transferred to an entirely different object that has never been seen before. Furthermore, it is important to note that the sensory readings for this object-action pair is more difficult to detect than that of drawer opening. As described in the previous chapter, sensory modalities can either be discrete or continuous. With the drawers, all of the modalities are continuous (*e.g.* visually the drawer is increasing in size, the force steadily increases once the end is reached, the drawer slowly scrapes along when moving). Continuous values are easier to model; discrete signals are short so the models have only a short time window to capture any change. Furthermore, discrete signals are harder to distinguish from noise. Unlike the drawer, the lamp has two discrete changes that must be modeled: the audio and haptic signal when the lamp switch clicks. The only continuous signal is the visual change in light.

This situation is also slightly different from the drawer in that the lamp will not turn on if the algorithm stops prior to the ground truth location. By contrast, there are few consequences, in the case of the drawer, if we stop before it has reached the full 9 inches. In fact, it can be argued that stopping slightly sooner is better than stopping too late. However, for the lamps, if the robot stops too soon, this creates an event where the robot does not achieve the task of turning on the lamp. We can see in Figure 7.11 that there are several instances where the robot stops too early and would have failed in deciding when it had succeeded at turning on the lamp. We can also see an example of the robot not stopping even though the robot had reached the point past when the lamp would have toppled over (*i.e.* visual and audio condition). For the example seen in Figure 7.11, haptic and visual information are critical to the lamp although audio also plays an important role in stopping.

Similar to the drawer, the absolute distance in time between the ground truth and stopping point for the different lamps can be found in Table 7.8, Table 7.9, and Table 7.10. We

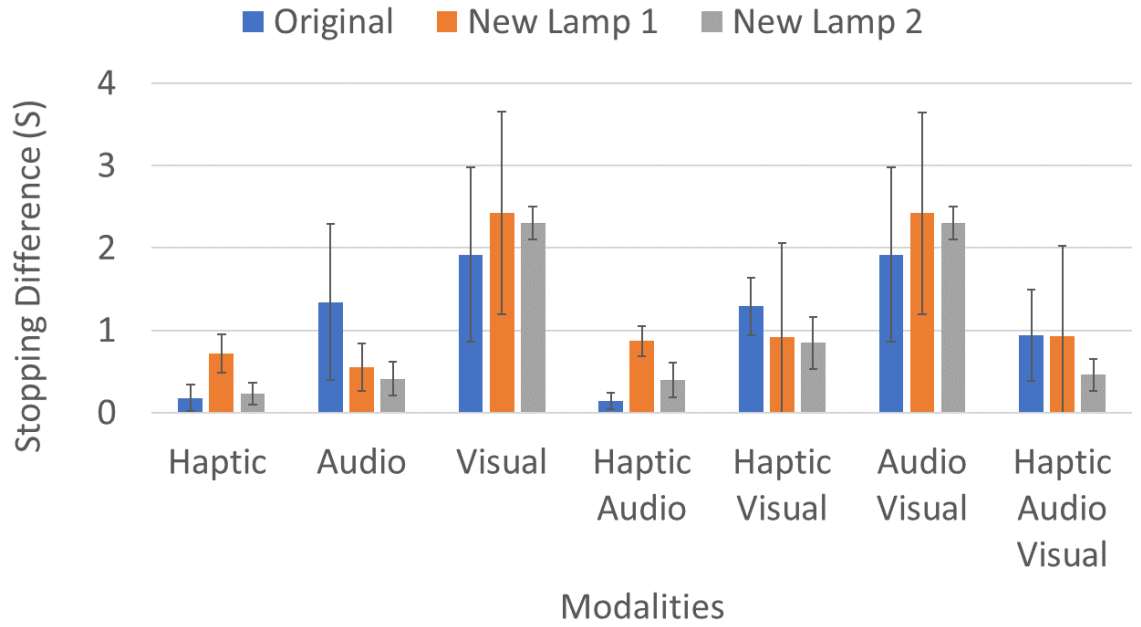


Figure 7.12: Average and standard deviation of absolute difference in stopping times for different lamps across 7 modalities.

Table 7.8: Original Lamp Detailed Times

Combo	1	2	3	4	5	Average	Success
Haptic	0.33	0.36	0.02	0.04	0.13	0.18 ± 0.16	4
Audio	0.24	2.04	0.64	2.52	1.27	1.34 ± 0.95	1
Visual	0.06	2.04	2.57	2.52	2.41	1.92 ± 1.06	1
Haptic Audio	0.22	0.26	0.02	0.11	0.08	0.14 ± 0.10	4
Haptic Visual	1.32	1.51	1.69	1.17	0.77	1.29 ± 0.35	3
Audio Visual	0.06	2.04	2.57	2.52	2.41	1.92 ± 1.06	1
Haptic Audio Visual	1.16	0.87	1.76	0.56	0.33	0.94 ± 0.56	3

Table 7.9: New Lamp 1 Detailed Times

Combo	1	2	3	4	5	Average	Success
Haptic	0.69	0.87	0.42	0.61	1.03	0.72 ± 0.24	0
Audio	0.49	0.48	0.18	0.97	0.64	0.55 ± 0.29	1
Visual	2.77	3.00	0.24	2.87	3.22	2.42 ± 1.23	1
Haptic Audio	0.85	0.91	0.70	0.75	1.16	0.87 ± 0.18	0
Haptic Visual	0.11	0.12	0.92	2.87	0.60	0.92 ± 1.14	3
Audio Visual	2.77	3.00	0.26	2.87	3.22	2.42 ± 1.22	0
Haptic Audio Visual	0.49	0.48	0.18	2.87	0.64	0.93 ± 1.10	1

Table 7.10: New Lamp 2 Detailed Times

Combo	1	2	3	4	5	Average	Success
Haptic	0.40	0.10	0.16	0.34	0.16	0.23 ± 0.13	4
Audio	0.18	0.36	0.51	0.30	0.71	0.41 ± 0.20	0
Visual	2.08	2.34	2.38	2.12	2.56	2.30 ± 0.20	0
Haptic Audio	0.18	0.34	0.51	0.26	0.71	0.40 ± 0.21	0
Haptic Visual	0.80	0.74	0.86	1.36	0.50	0.85 ± 0.32	5
Audio Visual	2.08	2.34	2.38	2.12	2.56	2.30 ± 0.20	0
Haptic Audio Visual	0.44	0.38	0.52	0.74	0.20	0.46 ± 0.20	5

add an additional metric in the final column to represent how many runs out of the 5 pulled far enough past the ground truth stopping point. We can see in the test case of the original lamp that the haptic channel is most successful in determining when to stop and has the lowest times. This is interesting because the haptic data for the lamp turning on is a discrete signal. However, if we look at Figure 7.7, we can see that there is a distinct increase in force that is felt that over time until the light turns on. This shows that the haptic signals are not just discrete, but also continuous leading up to the moment the lamp turns on.

If we look at the Figure 7.12, we can see that this trend roughly holds true across all three test cases. Similar to the drawer situation, the full sensory model (visual, haptic, audio) performs on par with the models that use a smaller subset of modalities (*e.g.* only haptic or haptic and audio). This shows the algorithm can automatically select the modalities that are important to the task without having an expert provide this information. Finally, we can see that overall, the models perform well but not perfectly at predicting when to stop pulling on the lamp. This is expected as this task requires much higher accuracy to successfully turn on the lamp as opposed to opening the drawer.

7.3 Online Validation: Adapting Learned Affordance Models in Real-time

For our final evaluation, we implement RMAC on Prentice and validate the offline results.

7.3.1 Real-time Controller

For this section, we use the controller described in Section 7.1.5 that was used to collect the offline test set described in Section 7.2.5. However, for this evaluation, we connect the data streams to a real-time feature extractor and connect it to the sensory models trained as described in Section 7.2.4. Specifically, we load the trained HMMs using all modalities (haptic, visual, and audio), and during the adaptive subskill segment playback, determine whether the robot has completed the particular subskill segment or if it should continue executing at the velocity v_n . The specifics of the real-time controller are broken down into the follow components:

1. **State Tracking:** The controller constantly tracks the current subskill segment.
2. **Plan Generation:** The controller loads the specified switching matrix S for the particular object-action model. At the beginning of each subskill segment, the controller generates a motion trajectory for the current subskill segment by either using the end-effector pose relative to the object (pose mode) or the v_n of the subskill trajectory (adaptive mode).
3. **Real-time Data Collection:** We create a real-time data collector that merges the different sensory streams into a single feature vector. This is implemented using a message filter built into ROS (Quigley et al., 2009) that triggers a callback when the incoming data streams are within a time range. In this work, that time range is set to 0.1 seconds. The data streams occur at different sample rates as seen in Table 7.1. The data is either down-sampled (audio) or up-sampled (visual) to match 100 Hz when the callback is triggered. The data is then stored in a buffer with the current subskill segment number.
4. **Action Execution:** Once the controller is loaded and has generated the plan for the current subskill, the robot executes either a long plan (pose mode) or a very short

trajectory created by v_n . If the controller is in an adaptive segment, the robot completes the small motion and queries the data collector to predict if it should continue or move on to the next subskill segment. When querying, the following two steps (Steps 5, 6) are executed.

5. **Feature Computation:** When requested, the controller selects all stored data streams from the current state and computes the 13 different features specified in Section 7.2.3. The features are normalized by removing the same mean and scaling factor computed during training.
6. **Prediction:** Once all of the features for the current subskill segment are generated, they are fed into the current subskill segment’s sensory model. The model determines the most likely path (Viterbi) through the left-to-right HMM given the feature vector over time. If the likely path reaches the final state of the HMM, the predictor returns stop. If the last state is not reached, the predictor returns continue.

7.3.2 Results

To replicate the offline results using the full multimodal model we execute five trials under each condition of the drawer (i.e. closed, 1in, 3in, 5in, 7in). During these trials, the robot executes the real-time controller to determine when to stop pulling on the drawer. To measure how successful the robot was at stopping at the right position for the drawer to be open, we measure two things: how far the drawer is open (9 inches is fully open) and how far the robot dragged the drawer set. The first metric tells us if the robot stopped too early (*i.e.* if the drawer is open less than 9 inches) and the second metric tells us if the robot stopped too late (*i.e.* if the robot starts to drag the drawer set across the table because it has not realized it should stop pulling). The average values across the 5 trials can be seen in Table 7.11. We can see in the online case that as the configuration gets more difficult (*i.e.* the controller has a smaller amount of time to make a decision), the distance the robot pulls the

Table 7.11: Online Drawer Results Using Full Multisensory Model

Configuration	Average Movement	Open Size	Non-Adapt Movement
0 inch	0.05 in	8.3 in	2 in
1 inch	0.08 in	8.36 in	3 in
3 inch	0.04 in	8.48 in	5 in
5 inch	0.16 in	9.0 in	7 in
7 inch	0.25 in	8.8 in	9 in

Table 7.12: Online Lamp Results Using Full Multisensory Model

Object	Accuracy
Orig Lamp	0.7
New Lamp 1	0.6
New Lamp 2	0.9

drawer increases. However, we can also see that if the robot did not perform adaptation, the distance pulled would be much greater.

The robot executed the controller 10 times for each online lamp evaluations. We report the accuracy of these interactions in Table 7.12. For the Original Lamp and New Lamp 2, the online results are fairly similar to the offline results. Interestingly, New Lamp 1 performs better than the offline results suggested. We believe this is related to the slight time delay between the real-time signals and the controller stopping. In the offline evaluation, we measured the exact moment the algorithm chooses to stop. However, in the real-time controller, there can exist a slight delay (up to one v_n motion). This appears to be enough to increase the success rate for the New Lamp 1 object.

We can see with the online evaluations that the controllers using the full multisensory model can be executed in real-time with results similar to those found offline. Furthermore, this evaluation gives a sense of scale to the offline evaluations. While some test configurations did not perform perfectly (*e.g.* the absolute difference in expected time and ground truth were greater than 0), this does not translate into large errors in real-time. In particular, we can see that the robot opens the drawer fully in all cases (*i.e.* between 8.3 and 9 inches) with only a few instances where the robot pulled slightly too long, which only translates to

dragging the drawer no greater than 0.25 inches. For the lamp evaluation, the difference in expected and ground truth stopping points is also small enough to not significantly impact the overall success rate of turning on the lamp.

7.4 Findings of Multisensory Affordances for Adaptive Object Manipulation

In this chapter, we introduced a novel approach to learning and executing affordances - RMAC. Specifically, we show that affordances can be adapted in certain situations where the object state or object have changed, but the other components (*e.g.* action, effects) have not. Furthermore, we show that using multisensory input improves the quality of the adaptation. The work shows that RMAC can be used across two very different objects and can select the modalities that are important to each task automatically without an expert specifying those modalities in advance. Furthermore, RMAC allows a robot to learn and execute affordances without explicitly specifying any objective function.

We demonstrate RMAC in two separate evaluations (offline and online) across two very different objects (drawer and lamp). The evaluations show that the combination of using haptic, audio, and visual information with RMAC allows the robot to open a drawer at 5 different configurations and turn on two never-before-seen lamps. The real-time online evaluations verify the offline evaluations and show that RMAC allows the robot to accurately open different configurations of the drawer (within 1 inch of perfect) and can turn on novel lamps with an average accuracy 0.75%.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

Commercial robots available today may still struggle to operate in complex, human-centric environments but the algorithms and approaches described in this thesis will bring us closer to a world where that is no longer the case. Using human guidance and multisensory input in conjunction with a framework that models the world as affordances, robots can continuously learn how to manipulate objects in human-centric environments.

This thesis makes the following contributions to the field of robot learning: (1) a *human-centered framework for robot affordance learning* that demonstrates how human teachers can guide the robot in the modeling process throughout the entire pipeline of affordance learning; (2) several novel *human-guided robot self-exploration* algorithms that use human guidance to enable robots to efficiently explore the environment and learn affordance models for a diverse range of manipulation tasks; (3) a *multisensory affordance model* that integrates visual, haptic, and audio input; and (4) a novel control framework that allows *adaptive object manipulation using multisensory affordances*.

8.1 Human-Centered Framework for Robot Affordance Learning

Without a framework for robot affordance learning, the problem space can quickly become intractable. To address this challenge, we introduced a robot learning framework that used guidance from human teachers during various stages of the affordance modeling pipeline:

- Human-guided exploration allowed the robot to learn haptic affordances and build multisensory affordance models.
- Human-guided robot self-exploration - seeded by demonstrations from 10 naïve human teachers - allowed the robot to efficiently sample the infinitely large action space

for manipulating objects.

- Kinesthetic LfD using keyframes allowed the robot to quickly build action models of various object-action pairs.

This framework ultimately allowed us to teach robots five different affordances across 11 different objects and actions in a tractable manner.

8.2 Human-Guided Robot Self-Exploration

Human-guided robot self-exploration allowed robots to utilize human guidance (e.g. reduced time to explore the environment) while minimizing the costs (e.g. human effort required). We developed several novel algorithms that enabled robots to efficiently explore the environment without requiring a human teacher to constantly adjust the environment and compared the impact of three types of exploration on affordance-learning performance: (1) learning through self-exploration, (2) learning from supervised examples provided by 10 naïve users, and (3) a combined approach of self-exploration biased by user input. By analyzing aggregate performance of the teachers, we showed that a combined approach is the most efficient and successful (Chu et al., 2016b).

Additionally, through a deep analysis on individual human teachers (Chu and Thomaz, 2017), we concluded that teachers that provided demonstrations with a relatively limited set of ways to interact with the object resulted in data that could be used to create a model with consistently good recognition capabilities. By contrast, teachers whose guided exploration resulted in a large number of different ways to successfully interact with the object generated data that created a poorly-performing model.

8.3 Multisensory Affordance Model

Robots need the ability to integrate and use multisensory input to more robustly interact with the world. Multisensory affordance models provide robots with a more robust under-

standing of their environment.

Through careful study of multisensory (*i.e.* haptic, visual, and audio) input for robot manipulation, we built a unified representation for object manipulation using affordances, including the insight that not only can sensory input vary between continuous and discrete signals but also that this difference can impact the overall quality of an affordance model (Chu et al., 2016a). Furthermore, modeling multiple sensory modalities using HMMs improved the overall accuracy of adaptive object manipulation and allowed the robot to automatically determine what modalities to focus on for each object-action pair.

8.4 Adaptive Object Manipulation using Multisensory Affordances

Much in the same way that humans build upon previously learned skills to adapt to a new situation, robots need to learn this ability if they are to be effective in human-centric environments. This thesis tackled the affordance learning problem of how to transfer a previously learned object-action pair to a novel configuration or object.

We developed a novel approach to adapting object-action pairs by using segmentation and hybrid control: Real-time Multisensory Affordance-based Control (RMAC). We demonstrated that RMAC can be used across two very different objects without an expert having to specify the differences in advance. Furthermore, we showed that the combination of using haptic, audio, and visual information with RMAC allowed the robot to open a drawer in five different configurations and turn on two never-before-seen lamps.

8.5 Discussion and Future Work

There are several insights gained from the contributions of this thesis that lead to interesting future questions. We discuss them in this section.

8.5.1 Human Guidance

Throughout this thesis, we utilize human input - in the form of demonstrations and environmental scaffolding - to help the robot learn affordances. We show that this input plays a key role in allowing robots to interact with the environment. In Chapter 5, we looked deeper into how these demonstrations can be provided by non-experts and showed that naïve human teachers could provide primitive actions to manipulate several objects. We also gained insight into how individuals can differ significantly when providing demonstrations. We concluded that this variation was not successfully captured in the models and future work needs to address this limitation by looking into approaches that can capture the various ways robots can execute skills.

In Chapter 5, we limit the demonstrations to *pick* and *move*. This provided a concrete set of subgoals that the human teachers can provide. However, in both Chapter 4 and Chapter 6, we rely on demonstrations that are not limited in the number of keyframes. It remains an open question whether, given complete freedom in the number of keyframes for a demonstration, (1) a non-expert human teacher can provide demonstrations that correctly highlight the subgoals of an affordance and (2) the highlighted subgoals can be used to generate meaningful subskills for adaptive object manipulation. Future work in this area should look into whether naïve human teachers can provide open-ended keyframe demonstrations. Interestingly, as we do show that limiting demonstrations to specific subgoals can be taught by naïve teachers, we can also look at the question of whether we can break down all actions for affordances into a specific set of limited primitives.

This thesis focused on using keyframes as the primary input for kinesthetic teaching. However, it remains an open question whether other forms of LfD could be used to guide robots in learning affordances. LfD allows robots to quickly acquire actions to interact with the world. While Human-Guided Affordance Learning and RMAC do not solely rely on keyframes, future work should look at the viability of other LfD approaches that do not use keyframes. In particular, could other inputs provide the crucial subgoals of the affordance

that keyframes inherently provide? Additionally, this thesis assumes that demonstrations provided from teachers are high-quality. Future work should seek to understand how low-quality demonstrations could be enhanced or identified and whether the method of LfD impacts the quality of the demonstration.

While naïve human teachers were able to provide demonstrations for various objects in a brief period of time (*i.e.* 5 and 10 minutes), it is unclear how novelty plays a role in the teaching. In particular, over a longer period of time, could we expect people around a robot to continue to provide guidance in learning tasks? During the study in Chapter 5, users could cease teaching a particular affordance earlier than the time limit. We discovered that, for simpler affordances, some users got bored quickly and moved on before the allotted time had elapsed. For the more difficult interactions, users seemed more engaged and experimented with more techniques. In particular, users tackled the challenge of turning on a lamp by slowly adapting and changing their approach when the robot initially failed. However, it is unclear if, over time, users would get frustrated with these harder affordances. Future work should look at understanding how people’s demonstrations and willingness to teach robots change over multiple interactions during a long time period.

8.5.2 Multisensory Input

There are many challenges in dealing with modalities of varying frequencies; this thesis depends on carefully hand-selected features. Future work should look at how to best generate features from the different sensory spaces and account for the difference in frequency without human intervention.

Additionally, while multisensory information is important throughout the execution of the trajectory, this thesis does not explore the robustness of RMAC if it only receives a partial set of sensory cues. Specifically, how could the robot adapt to a situation where it turns on a lamp that does not make any noise? This situation highlights an important component of multisensory adaptation: the primary goal or objective of the robot. In the

case of turning on a lamp, the primary sensory signal is light. Even if the robot were to hear a click and feel the correct amount of pressure, if there is no light, then the primary objective of turning on a lamp has not been achieved. Conversely, if the robot does see light but the lamp makes no noise, it should be able to reason that it has nonetheless achieved the goal of turning on the lamp. Future work should explore this notion of primary and secondary sensory modalities and enhance RMAC by adding an additional layer (*i.e.* high-level policy) on top of RMAC to determine primary and secondary success and failure.

8.5.3 Tasks

Affordances provide the basic building blocks that enable a robot to understand how its actions can change the environment and thus complete tasks. For robots to complete tasks robustly, however, they need to plan: this includes understanding when to perform a specific type of transfer and how to break down a task into basic steps (*e.g.* to water a plant, the robot should fill a containable object before pouring the water out of that object over the plant). Future work should look at how a robot can connect to a database of prior knowledge, ground this knowledge base with the multisensory adaptable controllers learned in this thesis, and execute these controllers in a manner that allows for robust task completion.

8.5.4 Summary

Through novel frameworks, algorithms, and multisensory models, this thesis enables robots to more quickly and effectively complete tasks in human-centric environments while simultaneously requiring less human intervention. Building upon the technical contributions in this thesis (which itself was built upon a strong foundation of prior work), we can continue to progress toward a future where robots can effectively respond to any request, regardless of the environment, the teacher, or the complexity of the task.

Appendices

APPENDIX A

USER STUDY

- Experimenter protocol
- Post-study questionnaire
- Experimenter data collection form
- IRB consent form

Experiment

1. Object-Actions-Effects (Affordance)

	Object	Action	Effect	Low Affordance
1	Bread box lid	Move	Moves up	Lift-able
2	Pasta jar	Move	Slides	Push-able
3	Drawer unit	Move	Slides	Push-able
4	Drawer shelf	Pick	Slides out	Pull-able
5	Lamp string	Pick	Clicks down	Pull-able

Subject Protocol

1. Person comes in - Sign consent forms
2. Explain study
 - a. Looking for affordances on specific objects. I will walk you through the specific affordances that I want you to teach the robot
3. You will do this through KLfD
 - a. Play with the robot and the commands to get a feel of how to move the robot around
 - i. Get them to move the arm to 2-3 specific poses (need to test)
 1. Elbow up
 2. Turning the wrist
 3. Closing the hand
 4. Lifting
4. To do the experiment you need to play with objects
 - a. Here is a test object (lift jar) (insert)

System Protocol

1. Initial Startup
 - a. Startup robot - (sync time if needed)
 - b. Startup vision system
 - i. ASUS launch
 - ii. Filtering
 - iii. Object Tracking
 - c. Startup c6
 - i. Select object + action
 1. Practice object + Move
2. Run through practice of KLfD
 - a. Commands (Release, hold, go here, end here, close your hand)
3. Run through practice of actual primitive
 - a. Go here
 - b. End here
 - i. I need to press "Generate Primitive"
 - c. Show me what you learned
 - i. Joint level playback so that we don't have issues with IK and collisions
 - d. Play with the actual practice object
 - i. Insertion
 - ii. Lift jar lid
4. Setup system for actual experiment
 - a. Restart c6
 - i. Select the object+action
 - b. Start data logging
 - c. Start video recording
5. Walk through each interaction
 - a. Have one interaction where the object location does not change
 - i. ie just have people demonstrate on the object with the object set on the predefined markers
 - ii. Then let them move the object around
 1. Do we want to let them re-teach?
 - a. Might get the height wrong for certain objects?
 - b. Hand record which interaction was successful vs. failure
 - i. Have a pre-printed sheet to write this on

Subject Instructions

- Welcome to lab - this is Curi the robot
- Goal is to teach affordances
 - Define Affordances
 - Action + Object = Effect
 - We are teaching the robot about what the affordance looks like and feels like to the robot
- Action is a primitive or “action template”
 - Pick
 - Move
- To each the action we'll be using Kinesthetic Learning from Demonstration
 - Physically guide the robot through the action
- Practice
 - KLfD
 - Try to move such that palm is up
 - Try to move such that thumb up, thumb down, palm down
 - Play with the elbow
 - Try to move towards the body
 - Try to move away from the body
 - Open and close hands - release and hold arms
 - Overall
 - You'll provide one demonstration
 - Then adjust the environment until we can teach the robot about the affordance, whether that means success or fail
 - I'll be recording if the action on the object was a successful affordance or failed affordance
 - Insertion
 - Insertable using the move primitive
 - Lift
 - The lid is liftable using the pick primitive

Affordance Survey

*** 1. What strategy did you use when teaching affordances to Curi? How did you decide what to show the robot?**

Feel free to provide multiple strategies if you had different strategies for each object or primitive.

Done

Powered by



SurveyMonkey®

See how easy it is to [create a survey](#).

User	Object	Action	Demo Type		User	Object	Action	Demo Type
			Success/Fail					Success/Fail
Demo	Explore Count	Result	Notes		Demo	Explore Count	Result	Notes
0	0				0	0		
0	1				0	1		
0	2				0	2		
0	3				0	3		
0	4				0	4		
0	5				0	5		
1	0				1	0		
1	1				1	1		
1	2				1	2		
1	3				1	3		
1	4				1	4		
1	5				1	5		
2	0				2	0		
2	1				2	1		
2	2				2	2		
2	3				2	3		
2	4				2	4		
2	5				2	5		
3	0				3	0		
3	1				3	1		
3	2				3	2		
3	3				3	3		
3	4				3	4		
3	5				3	5		
4	0				4	0		
4	1				4	1		
4	2				4	2		
4	3				4	3		
4	4				4	4		
4	5				4	5		
5	0				5	0		
5	1				5	1		
5	2				5	2		
5	3				5	3		
5	4				5	4		
5	5				5	5		
6	0				6	0		
6	1				6	1		
6	2				6	2		
6	3				6	3		
6	4				6	4		
6	5				6	5		

Georgia Institute of Technology
Project Title: Learning from Demonstration for Cloud Robotics
Investigators: Andrea L. Thomaz
Consent title: Main 07/13v1
Research Consent Form

You are being asked to be a volunteer in a research study.

Purpose:

The purpose of this study is:

- Evaluate the effectiveness of robot technology that is designed to learn new things by interacting with a human partner and investigate human preferences in the design of such technology.
- We estimate that 300 people will participate in this research over the course of the project. We are including only adults 18 years and older in this study.

Procedures:

If you decide to be in this study, your part will involve:

- The study will take between 30 minutes and an hour.
- You will first be given an introduction by the experimenter.
 - If your experimental condition involves interacting with the robot the experimenter will explain the functionality of the robot technology that will be used in the study. A demonstration of the interaction will be given by the experimenter. Then you will be asked to interact with the robot technology to complete a task in the robot's workspace in our lab. In this interaction you might be asked to physically move the robot's arms in the workspace or to collaborate with the robot on an object-directed task.
 - Otherwise you will be asked to teach a similar task but in a remote setting, where you are located in a separate room from the robot and have to operate it with a web-based interface in order to complete the teaching task.
- We will collect audio and video of the session. These tapes will remain confidential, and will not be distributed in any way without your permission (there is a video release option at the bottom of this form). They will only be used for human-robot interaction analysis.
- After the interaction there will be a short survey about your experience.
- Finally you will have an exit interview with the experimenter where you can ask questions and learn more about the goals of this research.

Risks/Discomforts

The following risks/discomforts may occur as a result of your participation in this study:



- The risks involved are no greater than those involved in daily activities, such as operating a microwave or using a computer. Any physical interactions will occur while the robot is in a compliant mode. Our hardware is specially designed to be compliant for safe physical interactions and the robot's movements can be stopped by the participant. Additional precautions include an emergency stop button near the experimenter at all times, and conservative speeds of operation.
- The video recording from this study might be viewed by the members of our research team other than the experimenter.

Benefits

The following benefits are possible as a result of being in this study:

- The potential benefits of this research, to society at large, are in the pursuit of humanoid robots that are able to work alongside human partners in natural and intuitive ways. This could lead to robots in society that can help us solve problems in the domain of healthcare, eldercare, education, smart manufacturing, among others.
- The robots being used in this research are state-of-the-art humanoid robots. The primary benefit that individuals typically find in participating in these studies is that this is an opportunity to get hands on experience interacting with a sophisticated robot and learning more about robotics at Georgia Tech.

Compensation to You Participants will receive \$10/hour in compensation, or when applicable participants will instead receive extra credit in an associated class (e.g., in the past CS 3600 students have received 1pt of extra credit toward their final grade for participating in similar research studies). If a participant decides to withdraw early, they will receive \$5.00 (or similar fraction of extra credit) for each half hour of participation.

U.S. Tax Law requires a mandatory withholding of 30% for nonresident alien payments of any type. Your address and citizenship/visa status may be collected for compensation purposes only. This information will be shared only with the Georgia Tech department that issues compensation, if any, for your participation.

Confidentiality

The following procedures will be followed to keep your personal information confidential in this study: The data that is collected about you will be kept private to the extent allowed by law. To protect your privacy, your records will be kept under a code number rather than by name. Your records will be kept in locked files and only study staff will be allowed to look at them. Your name and any other fact that might point to you will not appear when results of this study are presented or published.



- The video/audio tapes collected as part of the evaluation will be stored on an external hard drive for the duration of the analysis period. Dr. Thomaz and the student researchers involved with the project will have access to this data for the purpose of analyzing the human-robot interaction. This project is funded in a collaborative research grant with Dr. Sonia Chernova at Worcester Polytechnic Institute (WPI). Data from our experiments at Georgia Tech may also be shared with the broader research team of Dr. Chernova and her student researchers. This data will be backed up on a server managed by the College of Computing's Technology Services Organization. Credentials will be required for access to the data, ensuring that only the appropriate researchers have access. *The sponsor organization, the National Science Foundation, may also review study records.*

To make sure that this research is being carried out in the proper way, the Georgia Institute of Technology IRB may review study records. The Office of Human Research Protections may also look at study records.

Costs to You There are no costs involved with participating in this study.

In Case of Injury/Harm

If you are injured as a result of being in this study, please contact Dr. Andrea Thomaz at telephone (404) 385-3365. Neither the Principal Investigator nor Georgia Institute of Technology have made provision for payment of costs associated with any injury resulting from participation in this study.

Subject Rights

- Your participation in this study is voluntary. You do not have to be in this study if you don't want to be.
- You have the right to change your mind and leave the study at any time without giving any reason, and without penalty.
- Any new information that may make you change your mind about being in this study will be given to you.
- You will be given a copy of this consent form to keep.
- You do not waive any of your legal rights by signing this consent form.

Questions about the Study or Your Rights as a Research Subject

- If you have any questions about the study, you may contact Dr. Andrea Thomaz at telephone (404) 385-3365.
- If you have any questions about your rights as a research subject, you may contact Ms. Melanie Clark, Georgia Institute of Technology at (404) 894-6942.

If you sign below, it means that you have read (or have had read to you) the information given in this consent form, and you would like to be a volunteer in this study.



Subject Name

Subject Signature

Date

Signature of Person Obtaining Consent

Date

Video Release

I consent to the use of video recordings from this study in conference presentations and research promotion videos to be displayed on a project webpage.

YES ____ NO ____

Initial_____



APPENDIX B

CODE BASE

The software packages written and used throughout this thesis can be found in several Github organizations.

- HLP-R Organization: HLP-R Github Codebase
- Other: Socially Intelligent Machines Lab Github Codebase and Personal Github

The packages are listed below with their respective organization as well as a high-level description of the package. Each package has its own README or documentation can be found in the specific documentation repository on Github.

B.1 HLP-R

- `hlpr_common`: repository that stores the launch scripts to bringup basic Prentice functionality
- `hlpr_documentation`: wiki that documents the full setup of Prentice and all of the packages that are part of the HLP-R organization.
- `hlpr_kinesthetic_teaching`: package that contains the kinesthetic interaction pipeline where voice commands are used to interact with the robot (*e.g.* release the arm, open/close the gripper, etc.). It also contains the keyframe playback package that generates and plans trajectories using MoveIt!
- `hlpr_lookat`: Package that controls pan/tilt unit on Prentice. Locations in robot task space can be translated into pan/tilt commands.

- `hlpr_manipulation`: API and core scripts to interface the trajectory controller of the Jaco2 arm with the MoveIt! planning interface.
- `hlpr_perception`: Point Cloud library that performs object segmentation and basic feature computation (*e.g.* color, volume, centroid)
- `hlpr_simulator`: Gazebo simulator repository that fully simulates Prentice and a simple room environment.

B.2 Other

- `data_logger_bag`: Complete library that provides API for easy data logging using rosbag and conversion of bag files into HDF5 format for processing.
- `sklearn_suite`: API and scripts for all machine learning related code

APPENDIX C

DATA

The data sets collected throughout this thesis can be found on Github. Each dataset is listed with individual READMEs about how and what data was collected: Dataset Github

REFERENCES

- Abu-Dakka, FaresJ., Bojan Nemec, JimmyA. Jrgensen, ThiusiusR. Savarimuthu, Norbert Krger, and Ale Ude (2015). “Adaptation of manipulation skills in physical contact with the environment to reference force profiles”. In: *Autonomous Robots* 39.2, pp. 199–217.
- Akgun, Baris and Andrea Thomaz (2015). “Simultaneously learning actions and goals from demonstration”. In: *Autonomous Robots*, pp. 1–17.
- (2016). “Simultaneously learning actions and goals from demonstration”. In: *Autonomous Robots* 40.2, pp. 211–227.
- Akgun, Baris, Maya Cakmak, Karl Jiang, and Andrea L Thomaz (2012a). “Keyframe-based learning from demonstration”. In: *International Journal of Social Robotics* 4, pp. 343–355.
- Akgun, Baris, Maya Cakmak, Karl Jiang, and Andrea L. Thomaz (2012b). “Keyframe-based Learning from Demonstration”. In: *International Journal of Social Robotics* 4.4, pp. 343–355.
- Argall, Brenna D., Sonia Chernova, Manuela Veloso, and Brett Browning (2009). “A survey of robot learning from demonstration”. In: *Robotics and Autonomous Systems* 57.5, pp. 469–483.
- Beeson, Patrick and Barrett Ames (2015). “TRAC-IK: An Open-Source Library for Improved Solving of Generic Inverse Kinematics”. In: *Proceedings of the IEEE RAS Humanoids Conference*. Seoul, Korea.
- Bhattacharjee, Tapomayukh, James M Rehg, and Charles C Kemp (2014). “Inferring Object Properties from Incidental Contact with a Tactile Sensing Forearm”. In: *arXiv preprint arXiv:1409.4972*.
- Bhattacharjee, Tapomayukh, Ariel Kapusta, James M Rehg, and Charles C Kemp (2013). “Rapid categorization of object properties from incidental contact with a tactile sensing robot arm”. In: *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 219–226.
- Billard, A., S. Calinon, R. Dillmann, and S. Schaal (2008). “Survey: Robot Programming by Demonstration”. In: *Handbook of Robotics*. Vol. chapter 59. MIT Press.

- Calinon, S., F. Guenter, and A. Billard (2007). “On Learning, Representing, and Generalizing a Task in a Humanoid Robot”. In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 37.2, pp. 286–298.
- Chebotar, Y., O. Kroemer, and J. Peters (2014). “Learning robot tactile sensing for object manipulation”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 3368–3375.
- Chemero, Anthony (2003). “An outline of a theory of affordances”. In: *Ecological psychology* 15.2, pp. 181–195.
- Chernova, Sonia and Andrea L Thomaz (2014). *Robot Learning from Human Teachers*. Vol. 8. 3. Morgan & Claypool Publishers, pp. 1–121. published.
- Chu, Vivian and Andrea L. Thomaz (2017). “Analyzing differences between teachers when learning object affordances via guided exploration”. In: *The International Journal of Robotics Research* 36.5-7, pp. 739–758.
- Chu, Vivian, Baris Akgun, and Andrea L. Thomaz (2016a). “Learning haptic affordances from demonstration and human-guided exploration”. In: *2016 IEEE Haptics Symposium (HAPTICS)*, pp. 119–125.
- Chu, Vivian, Tesca Fitzgerald, and Andrea L. Thomaz (2016b). “Learning object affordances by leveraging the combination of human-guidance and self-exploration”. In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 221–228.
- Chu, Vivian, Reymundo A Gutierrez, Sonia Chernova, and Andrea L Thomaz (2017). “The Role of Multisensory Data for Automatic Segmentation of Manipulation Skills”. In: *RSS Workshop on (Empirically) Data-Driven Robotic Manipulation*.
- Chu, Vivian, Ian McMahon, Lorenzo Riano, Craig G. McDonald, Qin He, Jorge Martinez Perez-Tejada, Michael Arrigo, Naomi Fitter, John C. Nappo, Trevor Darrell, and Katherine J. Kuchenbecker (2013). “Using robotic exploratory procedures to learn the meaning of haptic adjectives”. In: *International Conference on Robotics and Automation (ICRA)*, pp. 3048–3055.
- Ciocarlie, Matei, Kaijen Hsiao, Edward Gil Jones, Sachin Chitta, Radu Bogdan Rusu, and Ioan A Şucan (2014). “Towards reliable grasping and manipulation in household environments”. In: *Experimental Robotics*. Springer, pp. 241–252.
- Csibra, Gergely (2003). “Teleological and Referential Understanding of Action in Infancy”. In: *Philosophical Transactions: Biological Sciences* 358.1431, pp. 447–458.

- Dogar, Mehmet Remzi, Maya Cakmak, Emre Ugur, and Ero Sahin (2007). “From primitive behaviors to goal-directed behavior using affordances”. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 729–734.
- Dogar, Mehmet Remzi, Emre Ugur, Erol Sahin, and Maya Cakmak (2008). “Using learned affordances for robotic behavior development”. In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 3802–3807.
- Ernst, Marc O. and Heinrich H. Blthoff (2004). “Merging the senses into a robust percept”. In: *Trends in Cognitive Sciences* 8.4, pp. 162 –169.
- Fischler, Martin A. and Robert C. Bolles (1981). “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Commun. ACM* 24.6, pp. 381–395.
- Fishel, Jeremy A and Gerald E Loeb (2012). “Bayesian exploration for intelligent identification of textures”. In: *Frontiers in neurorobotics* 6.
- Fitzpatrick, Paul, Giorgio Metta, Lorenzo Natale, Sajit Rao, and Giulio Sandini (2003). “Learning about objects through action - initial steps towards artificial cognition”. In: *International Conference on Robotics and Automation (ICRA)*, pp. 3140–3145.
- Gams, A., B. Nemec, A.J. Ijspeert, and A. Ude (2014). “Coupling Movement Primitives: Interaction With the Environment and Bimanual Tasks”. In: *Robotics, IEEE Transactions on* 30.4, pp. 816–830.
- Gams, Andrej, Tadej Petri, Martin Do, Bojan Nemec, Jun Morimoto, Tamim Asfour, and Ale Ude (2015). “Adaptation and coaching of periodic motion primitives through physical and visual interaction”. In: *Robotics and Autonomous Systems*, pp. –.
- Gaver, William W. (1993). “What in the World Do We Hear?: An Ecological Approach to Auditory Event Perception”. In: *Ecological Psychology* 5.1, pp. 1–29.
- Gemici, Mevlana C. and Ashutosh Saxena (2014). “Learning haptic representation for manipulating deformable food objects”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 638–645.
- Gibson, Eleanor J (2003). “The world is so full of a number of things: On specification and perceptual learning”. In: *Ecological psychology* 15.4, pp. 283–287.
- Gibson, James J. (1977). “The concept of affordances”. In: *Perceiving, acting, and knowing*, pp. 67–82.

- Grollman, Daniel H. and Aude G. Billard (2011). “Donut as I do: Learning from failed demonstrations”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3804–3809.
- (2012). “Robot Learning from Failed Demonstrations”. In: *International Journal of Social Robotics* 4.4, pp. 331–342.
- Hangl, S., E. Ugur, S. Szedmak, and J. Piater (2016). “Robotic playing for hierarchical complex skill learning”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2799–2804.
- Hermans, Tucker, James M Rehg, and Aaron F Bobick (2013a). “Decoupling behavior, perception, and control for autonomous learning of affordances”. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 4989–4996.
- Hermans, Tucker, Fuxin Li, James M Rehg, and Aaron F Bobick (2013b). “Learning stable pushing locations”. In: *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pp. 1–7.
- Ijspeert, A.J., J. Nakanishi, and S. Schaal (2002). “Movement imitation with nonlinear dynamical systems in humanoid robots”. In: *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*. Vol. 2, pp. 1398–1403.
- Ivaldi, Serena, Sao Mai Nguyen, Natalia Lyubova, Alain Droniou, Vincent Padois, David Filliat, Pierre-Yves Oudeyer, and Olivier Sigaud (2014). “Object Learning Through Active Exploration”. In: *Autonomous Mental Development, IEEE Transactions on* 6.1, pp. 56–72.
- Ivaldi, Serena, Natalia Lyubova, Damien Gérardiaux-Viret, Alain Droniou, Salvatore M Anzalone, Mohamed Chetouani, David Filliat, and Olivier Sigaud (2012). “Perception and human interaction for developmental learning of objects and affordances”. In: *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pp. 248–254.
- Jain, Advait and Charles C. Kemp (2013). “Improving Robot Manipulation with Data-driven Object-centric Models of Everyday Forces”. In: *Autonomous Robots* 35.2-3, pp. 143–159.
- Jamone, L., E. Ugur, A. Cangelosi, L. Fadiga, A. Bernardino, J. Piater, and J. Santos-Victor (2017). “Affordances in psychology, neuroscience and robotics: a survey”. In: *IEEE Transactions on Cognitive and Developmental Systems* PP.99, pp. 1–1.
- Kalakrishnan, Mrinal, L. Righetti, Peter Pastor, and S. Schaal (2011). “Learning force control policies for compliant manipulation”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 4639–4644.

- Kappler, Daniel, Peter Pastor, Mrinal Kalakrishnan, Manuel Wuthrich, and Stefan Schaal (2015). “Data-Driven Online Decision Making for Autonomous Manipulation”. In: *Proceedings of Robotics: Science and Systems*. Rome, Italy.
- Katz, D., M. Kazemi, J.A. Bagnell, and A. Stentz (2013a). “Interactive segmentation, tracking, and kinematic modeling of unknown 3D articulated objects”. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 5003–5010.
- Katz, Dov, Arun Venkatraman, Moslem Kazemi, J. Andrew (Drew) Bagnell, and Anthony (Tony) Stentz (2013b). “Perceiving, Learning, and Exploiting Object Affordances for Autonomous Pile Manipulation”. In: *RSS Berlin*.
- Khatib, O. (1987). “A unified approach for motion and force control of robot manipulators: The operational space formulation”. In: *Robotics and Automation, IEEE Journal of* 3.1, pp. 43–53.
- Kober, Jens, Andreas Wilhelm, Erhan Oztop, and Jan Peters (2012). “Reinforcement learning to adjust parametrized motor primitives to new situations”. In: *Autonomous Robots* 33.4, pp. 361–379.
- Konidaris, George, Scott Kuindersma, Roderic Grupen, and Andrew Barto (2012). “Robot learning from demonstration by constructing skill trees”. In: *The International Journal of Robotics Research* 31.3, pp. 360–375.
- Koppula, Hema S and Ashutosh Saxena (2013). “Anticipating human activities using object affordances for reactive robotic response”. In: *RSS, Berlin*.
- Kormushev, Petar, Sylvain Calinon, and Darwin G. Caldwell (2011a). “Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input”. In: *Advanced Robotics* 25.5, pp. 581–603.
- Kormushev, Petar, Sylvain Calinon, and Darwin G Caldwell (2011b). “Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input”. In: *Advanced Robotics* 25.5, pp. 581–603.
- Krishna, Ranjay, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei (2016). “Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations”. In:
- Kroemer, O., H. van Hoof, G. Neumann, and J. Peters (2014). “Learning to predict phases of manipulation tasks as hidden states”. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 4009–4014.

- Kroemer, O., C. Daniel, G. Neumann, H. van Hoof, and J. Peters (2015). “Towards learning hierarchical skills for multi-phase manipulation tasks”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 1503–1510.
- Krüger, Norbert, Christopher Geib, Justus Piater, Ronald Petrick, Mark Steedman, Florentin Wörgötter, Aleš Ude, Tamim Asfour, Dirk Kraft, Damir Omrčen, Alejandro Agostini, and Rüdiger Dillmann (2011). “Object–Action Complexes: Grounded abstractions of sensory–motor processes”. In: *Robotics and Autonomous Systems* 59.10, pp. 740–757.
- Kruger, V., D. Herzog, S. Baby, A. Ude, and D. Kragic (2010). “Learning Actions from Observations”. In: *Robotics Automation Magazine, IEEE* 17.2, pp. 30–43.
- Kuffner, J. J. and S. M. LaValle (2000). “RRT-connect: An efficient approach to single-query path planning”. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. Vol. 2, 995–1001 vol.2.
- Lederman, Susan J and Roberta L Klatzky (1993). “Extracting object properties through haptic exploration”. In: *Acta psychologica* 84.1, pp. 29–40.
- Li, Qiang, Christof Elbrechter, Robert Haschke, and Helge Ritter (2013). “Integrating vision, haptics and proprioception into a feedback controller for in-hand manipulation of unknown objects”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2466–2471.
- Lopes, Manuel, Francisco S Melo, and Luis Montesano (2007). “Affordance-based imitation learning in robots”. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1015–1021.
- Lynott, Dermot and Louise Connell (2009). “Modality exclusivity norms for 423 object properties”. In: *Behavior Research Methods* 41.2, pp. 558–564.
- Manschitz, Simon, Jens Kober, Michael Gienger, and Jan Peters (2015). “Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations”. In: *Robotics and Autonomous Systems* 74, Part A, pp. 97 –107.
- Martin-Martin, R. and O. Brock (2014). “Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 2494–2501.
- Mascolo, Michael F (2005). “Change processes in development: The concept of coactive scaffolding”. In: *New Ideas in Psychology* 23.3, pp. 185–196.

- Mason, Matthew T. (1981). “Compliance and Force Control for Computer Controlled Manipulators”. In: *Systems, Man and Cybernetics, IEEE Transactions on* 11.6, pp. 418–432.
- McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto (2015). “librosa: Audio and music signal analysis in python.” In: *14th python in science conference*, pp. 18–25.
- McLeod, Saul (2010). *Zone of Proximal Development*. (Visited on 07/13/2017).
- Meltzoff, Andrew N. and Jean Decety (2003). “What imitation tells us about social cognition: a rapprochement between developmental psychology and cognitive neuroscience”. In: *Philosophical Transactions of The Royal Society B: Biological Sciences* 358 (1431), pp. 491–500.
- Min, H., C. Yi, R. Luo, J. Zhu, and S. Bi (2016). “Affordance Research in Developmental Robotics: A Survey”. In: *IEEE Transactions on Cognitive and Developmental Systems* 8.4, pp. 237–255.
- Mohseni-Kabir, Anahita (2015). “Collaborative learning of hierarchical task networks from demonstration and instruction”. PhD thesis. Worcester Polytechnic Institute.
- Moldovan, Bogdan, Plinio Moreno, Martijn van Otterlo, José Santos-Victor, and Luc De Raedt (2012). “Learning relational affordance models for robots in multi-object manipulation tasks”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 4373–4378.
- Montesano, Luis, Manuel Lopes, Alexandre Bernardino, and José Santos-Victor (2008). “Learning object affordances: From sensory–motor coordination to imitation”. In: *Transactions on Robotics* 24.1, pp. 15–26.
- National Science Foundation, NSF (2014). *The realization of co-robots acting in direct support of individual and groups*.
- Nguyen, Hai and Charles C. Kemp (2014). “Autonomously learning to visually detect where manipulation will succeed”. In: *Autonomous Robots* 36.1, pp. 137–152.
- Nguyen, Sao Mai and Pierre-Yves Oudeyer (2014). “Socially guided intrinsic motivation for robot learning of motor skills”. In: *Autonomous Robots* 36.3, pp. 273–294.
- Niekum, S., S. Osentoski, C.G. Atkeson, and A.G. Barto (2015a). “Online Bayesian change-point detection for articulated motion models”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 1468–1475.

- Niekum, Scott, Sarah Osentoski, George Konidaris, Sachin Chitta, Bhaskara Marthi, and Andrew G. Barto (2015b). “Learning grounded finite-state representations from unstructured demonstrations”. In: *The International Journal of Robotics Research* 34.2, pp. 131–157.
- Oudeyer, Pierre-Yves, Frdric Kaplan, and Verena V Hafner (2007a). “Intrinsic Motivation Systems for Autonomous Mental Development”. In: *Evolutionary Computation, IEEE Transactions on* 11.2, pp. 265–286.
- (2007b). “Intrinsic Motivation Systems for Autonomous Mental Development”. In: *Evolutionary Computation, IEEE Transactions on* 11.2, pp. 265–286.
- Park, D., Z. Erickson, T. Bhattacharjee, and C. C. Kemp (2016). “Multimodal execution monitoring for anomaly detection during robot manipulation”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 407–414.
- Pastor, P., M. Kalakrishnan, L. Righetti, and S. Schaal (2012a). “Towards Associative Skill Memories”. In: *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pp. 309–315.
- Pastor, Peter, H. Hoffmann, T. Asfour, and S. Schaal (2009a). “Learning and generalization of motor skills by learning from demonstration”. In: *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 763–768.
- (2009b). “Learning and generalization of motor skills by learning from demonstration”. In: *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 763–768.
- Pastor, Peter, L. Righetti, Mrinal Kalakrishnan, and S. Schaal (2011a). “Online movement adaptation based on previous sensor experiences”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 365–371.
- (2011b). “Online movement adaptation based on previous sensor experiences”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 365–371.
- Pastor, Peter, Mrinal Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal (2011c). “Skill learning and task outcome prediction for manipulation”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3828–3834.
- (2011d). “Skill learning and task outcome prediction for manipulation”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3828–3834.

- Pastor, Peter, Mrinal Kalakrishnan, Ludovic Righetti, and Stefan Schaal (2012b). “Towards Associative Skill Memories”. In: *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pp. 309–315.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.
- Petric, T., A. Gams, L. Zlajpah, and A. Ude (2014). “Online learning of task-specific dynamics for periodic tasks”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 1790–1795.
- Pillai, Sudeep, Matthew Walter, and Seth Teller (2014). “Learning Articulated Motions From Visual Demonstration”. In: *Proceedings of Robotics: Science and Systems*. Berkeley, USA.
- Power, Thomas G (2000). *Play and exploration in children and animals*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Quigley, Morgan, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng (2009). “ROS: an open-source Robot Operating System”. In: *ICRA Workshop on Open Source Software*.
- Rabiner, Lawrence and Biing-Hwang Juang (1986). “An introduction to hidden Markov models”. In: *ASSP Magazine, IEEE* 3.1, pp. 4–16.
- Raibert, Marc H and John J Craig (1981). “Hybrid position/force control of manipulators”. In: *Journal of Dynamic Systems, Measurement, and Control* 103.2, pp. 126–133.
- Righetti, Ludovic, Mrinal Kalakrishnan, Peter Pastor, Jonathan Binney, Jonathan Kelly, RandolphC. Voorhies, GauravS. Sukhatme, and Stefan Schaal (2014). “An autonomous manipulation system based on force control and optimization”. In: *Autonomous Robots* 36.1-2, pp. 11–30.
- Romano, Joseph M, Kaijen Hsiao, Günter Niemeyer, Sachin Chitta, and Katherine J Kuchenbecker (2011). “Human-Inspired Robotic Grasp Control With Tactile Sensing”. In: *IEEE Transactions on Robotics* 27.6, pp. 1067–1079.
- Romano, Joseph M., Jordan P. Brindza, and Katherine J. Kuchenbecker (2013). “ROS open-source audio recognizer: ROAR environmental sound detection tools for robot programming”. In: *Autonomous Robots* 34.3, pp. 207–215.

- Rozo, L., P. Jimenez, and C. Torras (2013a). “Force-based robot learning of pouring skills using parametric hidden Markov models”. In: *Robot Motion and Control (RoMoCo), 2013 9th Workshop on*, pp. 227–232.
- (2011). “Robot learning from demonstration of force-based tasks with multiple solution trajectories”. In: *Advanced Robotics (ICAR), 2011 15th International Conference on*, pp. 124–129.
- Rozo, Leonel, Pablo Jimenez, and Carme Torras (2013b). “A robot learning from demonstration framework to perform force-based manipulation tasks”. In: *Intelligent Service Robotics* 6.1, pp. 33–51.
- (2013c). “A robot learning from demonstration framework to perform force-based manipulation tasks”. In: *Intelligent Service Robotics* 6.1, pp. 33–51.
- Şahin, Erol, Maya Çakmak, Mehmet R Doğar, Emre Uğur, and Göktürk Üçoluk (2007). “To Afford or Not to Afford: A New Formalization of Affordances Toward Affordance-Based Robot Control”. In: *Adaptive Behavior* 15.4, pp. 447–472.
- Schaal, Stefan, Auke Ijspeert, and Aude Billard (2003). “Computational approaches to motor learning by imitation”. In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 358.1431, pp. 537–547.
- Schmidhuber, Juergen (1991). “Curious model-building control systems”. In: *Neural Networks, 1991. 1991 IEEE International Joint Conference on*, 1458–1463 vol.2.
- Silva, Bruno Castro da, George Konidaris, and Andrew G. Barto (2014). “Active Learning of Parameterized Skills”. In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1737–1745.
- Sinapov, Jivko (2013). “Behavior-grounded multi-sensory object perception and exploration by a humanoid robot”. PhD thesis. Iowa State University.
- Sinapov, Jivko, Connor Schenck, Kerrick Staley, Vladimir Sukhoy, and Alexander Stoytchev (2014a). “Grounding semantic categories in behavioral interactions: Experiments with 100 objects”. In: *Robotics and Autonomous Systems* 62.5. Special Issue Semantic Perception, Mapping and Exploration, pp. 632–645.
- Sinapov, Jivko, Mark Wiemer, and Alexander Stoytchev (2009). “Interactive learning of the acoustic properties of household objects”. In: *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 2518–2524.
- Sinapov, Jivko, Taylor Bergquist, Connor Schenck, Ugonna Ohiri, Shane Griffith, and Alexander Stoytchev (2011a). “Interactive object recognition using proprioceptive and

- auditory feedback”. In: *The International Journal of Robotics Research* 30.10, pp. 1250–1262.
- Sinapov, Jivko, Connor Schenck, and Alexander Stoytchev (2014b). “Learning relational object categories using behavioral exploration and multimodal perception”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5691–5698.
- Sinapov, Jivko, Vladimir Sukhoy, Ritika Sahai, and Alexander Stoytchev (2011b). “Vibrotactile Recognition and Categorization of Surfaces by a Humanoid Robot”. In: *IEEE Transactions on Robotics* 27.3, pp. 488–497.
- Stoytchev, Alexander (2005). “Behavior-Grounded Representation of Tool Affordances”. In: *International Conference on Robotics and Automation (ICRA)*, pp. 3060–3065.
- Sturm, Jürgen, Advait Jain, Cyrill Stachniss, Charles C Kemp, and Wolfram Burgard (2010). “Operating articulated objects based on experience”. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 2739–2744.
- Su, Zhe, Oliver Kroemer, Gerald E. Loeb, Gaurav S. Sukhatme, and Stefan Schaal (2016). “Learning to Switch Between Sensorimotor Primitives Using Multimodal Haptic Signals”. In: *From Animals to Animats 14: 14th International Conference on Simulation of Adaptive Behavior, SAB 2016, Aberystwyth, UK, August 23-26, 2016, Proceedings*. Ed. by Elio Tuci, Alexandros Giagkos, Myra Wilson, and John Hallam. Cham: Springer International Publishing, pp. 170–182.
- Sukhoy, Vladimir, Veselin Georgiev, Todd Wegter, Ramy Sweidan, and Alexander Stoytchev (2012). “Learning to slide a magnetic card through a card reader”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 2398–2404.
- Sung, Jaeyong, Seok Hyun Jin, and Ashutosh Saxena (2015). “Robobarista: Object Part based Transfer of Manipulation Trajectories from Crowd-sourcing in 3D Pointclouds”. In: *ISRR*. Vol. abs/1504.03071.
- Taylor, Matthew E. and Peter Stone (2009). “Transfer Learning for Reinforcement Learning Domains: A Survey”. In: *J. Mach. Learn. Res.* 10, pp. 1633–1685.
- Technology, Georgia Institute of, Carnegie Mellon University, Robotics Technology Consortium, University of Pennsylvania, University of Southern California, Stanford University, University of California Berkeley, University of Washington, and Massachusetts Institute of Technology (2013). *A roadmap for U.S. Robotics: From Internet to Robotics*.
- Thomaz, Andrea L and Maya Cakmak (2009). “Learning about objects with human teachers”. In: *International Conference on Human Robot Interaction (HRI)*, pp. 15–22.

- Torres-Jara, Eduardo, Lorenzo Natale, and Paul Fitzpatrick (2005). “Tapping into touch”. In:
- Trevor, Alexander J. B., Suat Gedikli, Radu Bogdan Rusu, and Henrik I. Christensen (2013). “Efficient Organized Point Cloud Segmentation with Connected Components”. In:
- Ugur, E. and J. Piater (2015). “Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2627–2633.
- (2017). “Emergent Structuring of Interdependent Affordance Learning Tasks Using Intrinsic Motivation and Empirical Feature Selection”. In: *IEEE Transactions on Cognitive and Developmental Systems* 9.4, pp. 328–340.
- Ugur, E., Y. Nagai, E. Sahin, and E. Oztop (2015a). “Staged Development of Robot Skills: Behavior Formation, Affordance Learning and Imitation with Motionese”. In: *IEEE Transactions on Autonomous Mental Development* 7.2, pp. 119–139.
- Ugur, Emre, Yukie Nagai, Hande Celikkanat, and Erhan Oztop (2015b). “Parental scaffolding as a bootstrapping mechanism for learning grasp affordances and imitation skills”. In: *Robotica* 33.5, 11631180.
- Varadarajan, Karthik Mahesh and Markus Vincze (2012). “AfRob: The affordance network ontology for robots”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1343–1350.
- Vigorito, Christopher M and Andrew G Barto (2010). “Intrinsically Motivated Hierarchical Skill Learning in Structured Environments”. In: *Autonomous Mental Development, IEEE Transactions on* 2.2, pp. 132–143.
- Vygotsky, Lev (1978). “Interaction between learning and development”. In: *Readings on the development of children* 23.3, pp. 34–41.
- Wang, C., K. V. Hindriks, and R. Babuska (2014). “Effective transfer learning of affordances for household robots”. In: *4th International Conference on Development and Learning and on Epigenetic Robotics*, pp. 469–475.
- Wieland, Steven, D Gonzalez-Aguirre, Nikolaus Vahrenkamp, Tamim Asfour, and Rüdiger Dillmann (2009). “Combining force and visual feedback for physical interaction tasks in humanoid robots”. In: *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pp. 439–446.

- Wilcox, Teresa, Rebecca J. Woods, Catherine Chapa, and Sarah McCurry (2007). “Multi-sensory exploration and object individuation in infancy.” In: *Developmental psychology* 43 2, pp. 479–95.
- Wood, David, Jerome S. Bruner, and Gail Ross (1976). “THE ROLE OF TUTORING IN PROBLEM SOLVING*.” In: *Journal of Child Psychology and Psychiatry* 17.2, pp. 89–100.
- Zech, Philipp, Simon Haller, Safoura Rezapour Lakani, Barry Ridge, Emre Ugur, and Justus Piater (2017). “Computational models of affordance in robotics: a taxonomy and systematic classification”. In: *Adaptive Behavior* 25.5, pp. 235–271.
- Zhu, Yuke, Alireza Fathi, and Li Fei-Fei (2014). “Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II”. In: ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing. Chap. Reasoning about Object Affordances in a Knowledge Base Representation, pp. 408–424.